

Школа: Инженерная школа информационных технологий и робототехники  
 Направление подготовки: Информационные системы и технологии  
 Отделение школы (НОЦ): Отделение информационных технологий

### БАКАЛАВРСКАЯ РАБОТА

<b>Тема работы</b> <b>Технология RPA применительно к автоматизации тестирования программного обеспечения</b>
---

УДК 004/455/1^004/054

Студент

Группа	ФИО	Подпись	Дата
8И7А	Смолякова Кристина Викторовна		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	Кандидат технических наук		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Маланина Вероника Анатольевна	Кандидат экономических наук		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент ООД	Аверкиев Алексей Анатольевич			

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	Кандидат технических наук		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП

Код компетенции	Наименование компетенции
<b>Универсальные компетенции</b>	
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной форме на государственном и иностранном (-ых) языке
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этническом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течении сей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
<b>Общепрофессиональные компетенции</b>	
ОПК(У)-1	Владеет широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий
ОПК(У)-2	Способен использовать основные законы естественнонаучных дисциплин в профессиональной деятельности, применять методы математического анализа и моделирования, теоретического и экспериментального исследования
ОПК(У)-3	Способен применять основные приемы и законы создания и чтения чертежей и документации по аппаратным и программным компонентам информационных систем
ОПК(У)-4	Понимает сущность и значения информации в развитии современного информационного общества, соблюдает основные требования к информационной безопасности, в том числе защите государственной тайны
ОПК(У)-5	Способен использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению
ОПК(У)-6	Способен выбирать и оценивать способ реализации информационных систем и устройств (программно-, аппаратно- или программно-аппаратно-) для решения поставленной задачи
<b>Профессиональные компетенции</b>	
ПК(У)-11	Способен к проектированию базовых и прикладных информационных технологий

<b>ПК(У)-12</b>	Способен разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)
<b>ПК(У)-13</b>	Способен разрабатывать средства автоматизированного проектирования информационных технологий
<b>ПК(У)-14</b>	Способен использовать знание основных закономерностей функционирования биосферы и принципов рационального природопользования для решения задач профессиональной деятельности
<b>ДПК(У)-1</b>	Способен использовать технологии разработки объектов профессиональной деятельности в бизнесе и осуществлять все виды деятельности в условиях экономики информационного общества.

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники  
 Направление подготовки: Информационные системы и технологии  
 Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Цапко И.В.  
 (Подпись)    (Дата)    (Ф.И.О.)

**ЗАДАНИЕ  
на выполнение выпускной квалификационной работы**

В форме:

бакалаврской работы
(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)

Студенту:

Группа	ФИО
8И7А	Смоляковой Кристине Викторовне

Тема работы:

<b>Технология RPA применительно к автоматизации тестирования программного обеспечения</b>	
Утверждена приказом директора (дата, номер)	05.02.2021 №36-82/с

Срок сдачи студентом выполненной работы:

	11.06.2021
--	------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b></p> <p><i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</i></p>	<p>Исследование возможности использования технологии RPA как инструмента для автоматизации различных видов тестирования программного обеспечения. А также проведению сравнительного анализа данной технологии с уже имеющимися средствами автоматизации.</p>
---	--

<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <i>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</i></p>	<ol style="list-style-type: none"> <li>1. Анализ предметной области;</li> <li>2. Проектирование и разработка автоматизированных тестовых сценариев;</li> <li>3. Анализ полученных результатов;</li> <li>4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение;</li> <li>5. Социальная ответственность.</li> </ol>
--	--

<p><b>Перечень графического материала</b>  <i>(с точным указанием обязательных чертежей)</i></p>	Презентация в формате *.pptx
--	------------------------------

<p><b>Консультанты по разделам выпускной квалификационной работы</b>  <i>(с указанием разделов)</i></p>	
<p style="text-align: center;"><b>Раздел</b></p>	<p style="text-align: center;"><b>Консультант</b></p>
<p>Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</p>	<p style="text-align: center;">Маланина Вероника Анатольевна</p>
<p>Социальная ответственность</p>	<p style="text-align: center;">Аверкиев Алексей Анатольевич</p>

<p><b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b></p>	<p style="text-align: center;">26.01.2021</p>
--	---

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<p style="text-align: center;">Доцент ОИТ</p>	<p style="text-align: center;">Цапко Ирина Валериевна</p>	<p style="text-align: center;">Кандидат технических наук</p>		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
<p style="text-align: center;">8И7А</p>	<p style="text-align: center;">Смолякова Кристина Викторовна</p>		

Министерство науки и высшего образования Российской Федерации  
 федеральное государственное автономное  
 образовательное учреждение высшего образования  
 «Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники

Направление подготовки: Информационные системы и технологии

Уровень образования: бакалавриат

Отделение школы (НОЦ): Отделение информационных технологий

Период выполнения: весенний семестр 2020/2021 учебного года

Форма представления работы:

бакалаврская работа
---------------------

(бакалаврская работа, дипломный проект/работа, магистерская диссертация)

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	11.06.2021
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
	Основная часть	75
	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
	Социальная ответственность	10

**СОСТАВИЛ:**

**Руководитель ВКР**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	Кандидат технических наук		

**СОГЛАСОВАНО:**

**Руководитель ООП**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ	Цапко Ирина Валериевна	Кандидат технических наук		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

Группа	ФИО
8И7А	Смоляковой Кристине Викторовне

Школа	ИШИТР	Отделение школы (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Оклад студента – 19 500 руб. Оклад руководителя – 58 500 руб. Оклад руководителя от компании – 58 500 руб.
2. Нормы и нормативы расходования ресурсов	Премиальный коэффициент 30%; Коэффициент доплат и надбавок 20%; Районный коэффициент 1,3; Коэффициент дополнительной заработной платы 13%; Накладные расходы 16%.
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	Коэффициент отчислений на уплату во внебюджетные фонды 30,2%

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Описание целевой аудитории разрабатываемой системы. Анализ конкурентных технических решений Проведение QuaD-анализа. Проведение SWOT-анализа.
2. Планирование и формирование бюджета научных исследований	Определение трудоемкости выполнения работ; Расчет материальных затрат НИИ; Основная и дополнительная зарплата исполнителей; Отчисления во внебюджетные фонды; Накладные расходы.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определение эффективности исследования.

**Перечень графического материала (с точным указанием обязательных чертежей):**

<ol style="list-style-type: none"> <li>1. Оценка конкурентоспособности технических решений</li> <li>2. Матрица SWOT</li> <li>3. Диаграмма Ганта</li> <li>4. Расчет бюджета НИ</li> <li>5. Оценка ресурсной, финансовой и экономической эффективности НИ</li> </ol>	
--	--

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН	Маланина В. А.	к.э.н., доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И7А	Смолякова Кристина Викторовна		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа	ФИО
8И7А	Смоляковой Кристине Викторовне

Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

Тема ВКР:

Технология RPA применительно к автоматизации тестирования программного обеспечения	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p>1. Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика, рабочая зона) и области его применения</p>	<p>Объектом исследования является технология RPA в качестве инструмента для автоматизации процесса тестирования. Рабочей зоной для использования данной технологии является автоматизированное рабочее место специалиста по автоматизации тестирования в офисном помещении. К области применения данного инструмента стоит отнести отдел тестирования компаний, занимающихся разработкой ПО.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p><b>1. Правовые и организационные вопросы обеспечения безопасности:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<ul style="list-style-type: none"> <li>– Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 09.03.2021);</li> <li>– ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования;</li> <li>– ГОСТ 21889-76. Система «человек-машина». Кресло человека-оператора. Общие эргономические требования;</li> <li>– ГОСТ 22269-76. Система «человек-машина». Рабочее место оператора. Взаимное расположение элементов рабочего места. Общие эргономические требования;</li> <li>– ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре;</li> <li>– ГОСТ Р ИСО 9241-5-2009. Эргономические требования к</li> </ul>

	<p>проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора.</p> <ul style="list-style-type: none"> <li>– ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности;</li> <li>– СП 51.13330.2011. Защита от шума. Актуализированная редакция СНиП 23-03-2003;</li> <li>– СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*;</li> <li>– ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности;</li> <li>– ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны;</li> <li>– ГОСТ 12.0.003-2015 ССБТ. Опасные и вредные производственные факторы. Классификация.</li> </ul>
<p><b>2. Производственная безопасность:</b>  2.1. Анализ выявленных вредных и опасных факторов  2.2. Обоснование мероприятий по снижению воздействия</p>	<ul style="list-style-type: none"> <li>– Повышенный уровень шума;</li> <li>– Недостаточная освещенность рабочей зоны;</li> <li>– Повышенный уровень электромагнитных излучений;</li> <li>– Отклонение показателей микроклимата.</li> </ul>
<p><b>3. Экологическая безопасность:</b></p>	<p>Литосфера, атмосфера и гидросфера: неправильная утилизация средств разработки программного обеспечения.</p>
<p><b>4. Безопасность в чрезвычайных ситуациях:</b></p>	<p>Возможные ЧС: пожар и взрыв, обрушение зданий, аварии на электроэнергетических системах, эпидемия, пандемия.  Наиболее типичная ЧС: пожар.</p>

Дата выдачи задания для раздела по линейному графику	
--	--

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Аверкиев Алексей Анатольевич	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8И7А	Смолякова Кристина Викторовна		

## РЕФЕРАТ

Выпускная квалификационная работа состоит из 85 с., 24 рис., 22 табл., 17 источников.

Ключевые слова: автоматизация, RPA, тестирование, технология, программное обеспечение.

Объект исследования: технология RPA в качестве инструмента для автоматизации процесса тестирования.

Цель работы: Исследовать возможность использования технологии RPA как инструмента для автоматизации различных видов тестирования программного обеспечения. А также проведению сравнительного анализа данной технологии с уже имеющимися средствами автоматизации.

В результате выполнения работы были проведены процессы тестирования с использованием различных инструментов и техник. По результатам чего был проведен сравнительный анализ применения технологии RPA с уже имеющимися средствами автоматизации. И рассчитана эффективность внедрения автоматизации в проект.

Область применения: отдел тестирования компаний, занимающихся разработкой ПО.

В будущем планируется расширить знания и практические навыки применения рассмотренных технологий в процессе тестирования программного обеспечения.

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

ПО – программное обеспечение.

RPA – Robotic process automation (Роботизированная автоматизация процессов).

ЖЦ – жизненный цикл.

BPМN - Business Process Model and Notation (модель бизнес-процессов и нотация).

ID – уникальный идентификатор.

ROI – return on investment (коэффициент окупаемости).

QA – quality assurance (обеспечение качества).

HTML – HyperText Markup Language (язык гипертекстовой разметки).

CSS – Cascading Style Sheets (язык таблиц стилей).

XPath – XML Path Language (язык запросов к элементам xml или xhtml документа).

## ОГЛАВЛЕНИЕ

Введение .....	14
1 Анализ предметной области .....	16
1.1 Жизненный цикл тестирования .....	16
1.2 Уровни и виды тестирования.....	18
1.3 Тестирование по степени автоматизации.....	20
1.3.1 Сравнение ручного и автоматизированного тестирования .....	20
1.3.2 Области применения ручного и автоматизированного тестирования .	24
2 Автоматизация тестирования с использованием Selenium .....	27
2.1 Используемые инструменты и технологии .....	27
2.1.1 Библиотека для написания автоматизированных тест-кейсов .....	28
2.1.2 Паттерн программирования для автоматизации .....	30
2.2 Процесс автоматизации тестовых сценариев.....	31
2.2.1 Разработка тест-кейсов для ручного тестирования.....	31
2.2.2 Разработка автоматизированных тест-кейсов .....	33
2.2.3 Выполнение автоматизированных тест-кейсов.....	37
3 Автоматизация тестирования с помощью RPA .....	38
3.1 UiPath Test Suite.....	40
3.1.1 UiPath Test Suite в жизненном цикле тестирования .....	41
3.2 Создание тестовых сценариев в StudioPro .....	42
3.2.1 Проверки .....	43
3.2.2 Тестовые сценарии.....	44
3.2.3 Тестовые сценарии на основе данных .....	48
3.3 Выполнение тестовых сценариев .....	50
4 Анализ полученных результатов .....	52
4.1 Сравнение инструментов автоматизации тестирования .....	52
4.2 Сравнение результатов проведения тестирования .....	53
4.3 Оценка эффективности внедрения автоматизации.....	54
5 Социальная ответственность .....	57
5.1 Введение.....	57
5.2 Правовые и организационные вопросы обеспечения безопасности.....	57
5.3 Производственная безопасность.....	59
5.3.1 Анализ опасных и вредных производственных факторов.....	60

5.3 Экологическая безопасность.....	64
5.4 Безопасность в чрезвычайных ситуациях .....	65
5.5 Вывод по разделу.....	66
6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение ....	68
6.1 Введение.....	68
6.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	68
6.2.1 Потенциальные потребители результатов исследования .....	68
6.2.2 Анализ конкурентных технических решений .....	69
6.2.3 Технология QuaD.....	70
6.2.4 SWOT-анализ .....	71
6.3 Планирование научно-исследовательских работ.....	72
6.3.1 Структура работ в рамках научного исследования.....	72
6.3.2 Определение трудоемкости выполнения работ .....	73
6.3.3 Разработка графика проведения научного исследования.....	74
6.4 Бюджет научно-технического исследования (НТИ).....	76
6.4.1 Расчет затрат на специальное оборудование.....	76
6.4.2 Расчет основной заработной платы .....	77
6.4.3 Расчет дополнительной заработной платы.....	79
6.4.4 Расчет отчислений во внебюджетные фонды .....	79
6.4.5 Расчет накладных расходов.....	80
6.4.6 Формирование бюджета затрат научно-исследовательского проекта..	81
6.5 Вывод по разделу.....	81
Заключение .....	82
Список используемых источников.....	83

## Введение

Программные системы стали неотъемлемой частью нашей повседневной жизни. Они помогают нам удовлетворить многие наши потребности и даже решить некоторые проблемы, с которыми мы сталкиваемся. Мы используем домашние банковские приложения, заказываем еду, планируем путешествия и делаем покупки. В нашей жизни не так уж много областей, которые так или иначе не затронуты программным обеспечением.

Эта эволюция произошла во всех отраслях. Программное обеспечение стало технологией номер один в любом бизнесе. Таким образом, программный сбой становится бизнес-неудачей. Именно поэтому тестирование программного обеспечения имеет первостепенное значение. Это снижает риск отказа, помогая обнаруживать дефекты до того, как продукт попадет к покупателю. И, в конце концов, это помогает создавать качественные продукты.

На данный момент ни один цикл разработки программного обеспечения не обходится без этапа тестирования, так как тестирование нацелено на обеспечение качества разрабатываемой системы. Под качеством программного обеспечения понимают совокупность характеристик, которые позволяют определить, удовлетворяет ли программа установленным требованиям и выполняет ли она то, что от нее ожидают. Основной задачей тестирования программного обеспечения является снижение стоимости разработки путем раннего обнаружения дефектов.

Раннее обнаружение дефектов и своевременная обратная связь позволяют создавать высококачественное и надежное программное обеспечение. Кроме того, организация эффективного и непрерывного тестирования ускоряет поставку этого программного обеспечения и снижает затраты компании по причине того, что у разработчиков появляется возможность вносить в код изменения с минимальными рисками, которые способны нарушить работоспособность программы.

Главное решение для непрерывного тестирования является его автоматизация. Автоматизация тестирования прошла долгий путь. Изначально

она внедрялась только для сокращения времени тестирования, сейчас к этому добавляется обеспечение оптимального тестового покрытия и более эффективное использование тест-кейсов. Таким образом, внедрение автоматизации тестирования преследует цель получения высококачественного программного продукта, и как можно быстрее.

Однако не всегда решение автоматизировать тестирование помогает решить вышеперечисленные задачи, да и само внедрение может привести к множеству проблем. Решение об использовании конкретного типа тестирования зависит в большей степени от задачи, решаемой на проекте. Так в данной работе необходимо определить области применения и виды тестирования, для которых эффективно автоматизированное тестирование, а также рассмотреть некоторые инструменты автоматизации.

## **1 Анализ предметной области**

Тестирование ПО – это процесс исследования и испытания программного продукта, имеющего две основные цели:

1. Проверка того, что программное обеспечение полностью соответствует требованиям и нуждам заказчика;
2. Выявление ситуаций, в которых поведение программы является неправильным.

Причем, проверка того, что программное обеспечение будет соответствовать нуждам заказчика является главной целью тестирования. Иначе, нет смысла проверять систему на правильность работы, если она изначально не будет соответствовать основным требованиям и нуждам заказчика.

Так как конечной целью тестирования является обеспечение качества продукта, то необходимо определить, что такое качество продукта.

Под качеством понимают степень, с которой компонент, система или процесс соответствует зафиксированным требованиям и ожидания, потребностям пользователя или заказчика.

### **1.1 Жизненный цикл тестирования**

Процесс тестирования имеет свой жизненный цикл, который включается в фазу тестирования в жизненном цикле разработки ПО. Не существует единого универсального подхода к процессу тестирования программного обеспечения. Каждый отдел тестирования может следовать определенному подходу в зависимости от нескольких факторов. Одним из таких факторов может быть подход к разработке, например, Waterfall или Agile.

Тем не менее, можно с уверенностью предположить, что типичный процесс тестирования проходит через следующие фазы: анализ требований, тестовое планирование, разработка тестовых случаев, выполнение тестов и анализ результатов выполнения тестов. Схема жизненного цикла тестирования ПО представлена на рисунке 1.



Рисунок 1 – Жизненный цикл тестирования ПО

Прежде чем перейти к описанию фаз жизненного цикла тестирования, необходимо сказать, что на любой фазе имеются входные параметры, без которых эта фаза не может начаться.

Фазы жизненного цикла тестирования:

1. Анализ требований. Входные параметры: требования. На данной фазе необходимо проанализировать требования на качества хороших требований, подготовить вопросы по требованиям, определить цели тестирования и приоритеты, а также подготовить тестовую среду.
2. Тестовое планирование. Входные параметры: откорректированные требования. На данной фазе происходит написание тестового плана, составление тестовой стратегии, а также оценка времени для тестирования.
3. Разработка тестовых случаев. Входные параметры: откорректированные требования. На данной фазе происходит написание ручных и автоматизированных тест-кейсов, подготовка тестовых данных и настройка тестовой среды.
4. Выполнение тестов. Входные параметры: готовые тест-кейсы, готовые тестовые данные, готовая тестовая среда, наличие ПО. На данном этапе происходит запуск ручных и автоматизированных тестовых наборов в соответствии с планом тестирования, нахождение и регистрация дефектов, и их проверка после исправления.

5. Анализ результатов выполнения тестов. Входные параметры: выполненные тест-кейсы, исправленные дефекты или исправлены самые критичные дефекты. На данном этапе происходит тщательный анализ результатов тестирования, подготовка и написание различных уровней отчетности о тестировании, что позволяет заинтересованным сторонам принять обоснованное решение о выпуске или доработке тестируемой системы.

Цель прохождения фаз жизненного цикла тестирования заключается в предоставлении подробной информации о качестве ПО, что позволит выявить риски, связанные с выпуском системы в эксплуатацию, намного раньше.

## **1.2 Уровни и виды тестирования**

Существует классификация видов тестирования, которая основана на том уровне детализации работ проекта, на который она нацелена. Эти же разновидности тестирования можно связать с фазой жизненного цикла, на которой они выполняются.

Модульное тестирование (Unit-testing) — уровень тестирования, на котором тестируется минимально возможный для тестирования компонент, например, отдельный класс или функция. На этом уровне применяются методы «белого ящика». В современных проектах модульное тестирование осуществляется разработчиками. Модульное тестирование предназначено для проверки правильности отдельных модулей, вне зависимости от их окружения. При этом проверяется, что если модуль получает на вход данные, удовлетворяющие определенным критериям корректности, то и результаты его корректны. Модульное тестирование является важной составной частью отладочного тестирования, выполняемого разработчиками для отладки написанного ими кода.

Интеграционное тестирование – уровень тестирования, на котором отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного

тестирования. Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных для последующего системного тестирования. Интеграционное тестирование выполняется разработчиками используется при отладке, но на более позднем этапе разработки.

Системное тестирование – это тестирование программного обеспечения, выполняемое на полной, интегрированной системе, с целью проверки соответствия системы исходным требованиям. Системное тестирование относится к методам тестирования «чёрного ящика» и не требует знаний о внутреннем устройстве системы. Системное тестирование выполняется через внешние интерфейсы программного обеспечения и тесно связано с тестированием пользовательского интерфейса, проводимым при помощи имитации действий пользователей над элементами этого интерфейса. Системное тестирование выполняется инженерами по тестированию.

Приемочное тестирование – это тестирование готового продукта конечными пользователями на реальном окружении, в котором будет функционировать тестируемое приложение. Приемочные тесты разрабатываются пользователями, обычно, в виде сценариев.

Регрессионное тестирование – тестирование функциональности, которая была уже протестирована до внесения какого-либо изменения. После внесения изменений в очередную версию программы, регрессионные тесты подтверждают, что сделанные изменения не повлияли на работоспособность остальной функциональности приложения. Регрессионное тестирование может выполняться как вручную, так и средствами автоматизации тестирования. Основная проблема регрессионного тестирования заключается в поиске компромисса между имеющимися ресурсами и необходимостью проведения таких тестов по мере внесения каждого изменения.

«Смок-тест» или дымовое тестирование означает минимальный набор тестов на явные ошибки. Не проходящую этот тест программу не имеет смысла отдавать на более глубокое тестирование.

### **1.3 Тестирование по степени автоматизации**

Одним из признаков в классификации тестирования является тестирование по степени автоматизации. Данный признак предполагает два подхода к тестированию – это ручное тестирование и автоматизированное тестирование. Нередко выделяют третий подход – это смешанное или полуавтоматизированное тестирование.

Ручное тестирование предполагает выполнение тест-кейсов без помощи каких-либо программ, автоматизирующих работу человека. То есть специалист по тестированию проверяет работоспособность системы вручную, как предполагаемый пользователь.

Автоматизированное тестирование предполагает исполнение тест-кейсов по заранее написанной программе специальным инструментальным средством. Несмотря на это разработка тест-кейсов, подготовка тестовых данных, анализ полученных результатов и составление различной отчетности выполняется вручную.

Смешанное или полуавтоматизированное тестирование сочетает в себе ручное и автоматизированное тестирование, где часть функциональности подвергается автоматизации, а другая часть выполняется вручную.

#### **1.3.1 Сравнение ручного и автоматизированного тестирования**

Автоматизация тестирования представляет набор техник, подходов и инструментальных средств, позволяющих исключить человека из выполнения некоторых задач в процессе тестирования. Несмотря на это любая автоматизация начинается с ручного тестирования, так как сначала тестирование необходимо четко спланировать и написать ручные тест-кейсы,

на основе которых будут разрабатываться автоматизированные тестовые сценарии.

Как ручное, так и автоматизированное тестирование имеют ряд преимуществ и недостатков, которые необходимо определить, чтобы подробнее разобраться в каких ситуациях следует применять автоматизированное тестирование, а в каких случаях эффективнее будет использовать ручное тестирование.

Анализ ручного и автоматизированного тестирования проводился по следующим признакам:

1. Скорость выполнения тест-кейсов. Автоматизация позволяет проводить тесты быстрее, чем если это делал бы человек. Особенно это ощущается при прохождении тестов, управляемых данными, так как необходимо проводить одни и те же проверки большое количество раз, но с разными наборами данных.
2. Возможность повторного использования. При автоматизированном тестировании написанные тестовые сценарии можно использовать в дальнейшем на проекте при его очередном обновлении сколько угодно раз и с наименьшими затратами по времени в отличие от проведения ручного тестирования.
3. Человеческий фактор. Так как некоторые проверки необходимо повторять большое количество раз, да и тестов на проекте может быть очень много, то можно утверждать, что человек гарантированно ошибется. Тогда как программа не допустит этих ошибок по невнимательности или неосторожности.
4. Выполнение тестирования производительности. Такие тесты могут быть непосильны человеку по ряду факторов, например, их сложность или скорость выполнения. Например, исследование производительности, когда необходимо с высокой скоростью выполнять определенные действия и фиксировать действия для большого объема параметров.

5. Квалификация персонала. Часто автоматизацию тестированию называют проектом внутри проекта, поэтому у людей, занимающихся автоматизацией тестирования, должна быть, техническая квалификация выше, чем у людей, занимающихся ручным тестированием.
6. Разработка и сопровождение тест-кейсов. Разработка автоматизированных тестовых сценариев занимает большее количество времени, а если в проекте происходит ситуации, когда вносятся ощутимые изменения, то тесты необходимо перерабатывать. Так на сопровождение тестов тоже уходит большое количество времени.
7. Планирование и управление рисками. Планирование и управление рисками при автоматизации тестирования крайне важно, так как может нанести огромный ущерб проекту как по времени, так и по бюджету.
8. Проведение исследовательского тестирования. Автоматизированные тесты выполняются строго по плану, в то время как при ручном тестировании можно обращать внимание на детали и можно найти неожиданные ошибки.
9. Стоимость внедрения. Большинство средств автоматизации стоят дорого, а бесплатные средства не всегда дают нужный эффект. Поэтому важно планировать внедрение автоматизации.
10. Человеческий взгляд. Возможно существование ошибок, которые может заметить только человек. Также существуют виды тестирования, которые способен выполнять только человек, например, тестирования удобства использования или тестирование требований.

Исходя из данных признаков были сформулированы основные достоинства и недостатки ручного тестирования и автоматизированного тестирования, которые представлены в таблице 1.

Таблица 1 – Сравнение ручного и автоматизированного тестирования

Автоматизированное тестирование	Ручное тестирование
Достоинства	
<ol style="list-style-type: none"> <li>1. Скорость выполнения тест-кейсов;</li> <li>2. Более раннее обнаружение дефектов;</li> <li>3. Возможность повторного использования;</li> <li>4. Отсутствует человеческий фактор;</li> <li>5. Экономия времени;</li> <li>6. Эффективность тестовых данных;</li> <li>7. Способность выполнять тест-кейсы непосильные для человека;</li> <li>8. Способность собирать, сохранять и анализировать большие объемы данных.</li> </ol>	<ol style="list-style-type: none"> <li>1. Дешевизна;</li> <li>2. Низкий порог входа специалистов;</li> <li>3. Проведение исследовательского тестирования;</li> <li>4. Тестирование удобства использования пользовательского интерфейса;</li> <li>5. Проверка документации.</li> </ol>
Недостатки	
<ol style="list-style-type: none"> <li>1. Наличие высококвалифицированного персонала;</li> <li>2. Разработка и сопровождение автоматизированных тест-кейсов и необходимой инфраструктуры;</li> <li>3. Необходимость планирования и управления рисками;</li> <li>4. Выполнение тестов строго по плану;</li> <li>5. Дороговизна;</li> <li>6. Отсутствие человеческого взгляда.</li> </ol>	<ol style="list-style-type: none"> <li>1. Низкая скорость проведения тестирования;</li> <li>2. Невозможно выполнять тестирование производительности;</li> <li>3. Влияние человеческого фактора;</li> <li>4. Трудоемкость повторного использования.</li> </ol>

Выявленные преимущества автоматизации тестирования позволяют увеличить тестовое покрытие за счет внедрения тестов, о которых раньше не могло идти речи, так как человек с ним не справился бы, за счет возможности многократно использовать тесты с разными входными данными и за счет освобождения ресурсов QA-инженера на создание новых тестов.

### 1.3.2 Области применения ручного и автоматизированного тестирования

Описанные в пункте 1.3.1 достоинства и недостатки ручного и автоматизированного тестирования позволяют выделить их области применения, которые представлены в таблице 2.

Таблица 2 – Области применения ручного и автоматизированного тестирования

Области применения	
Автоматизированное тестирование	Ручное тестирование
Регрессионное тестирование. Данное тестирование необходимо проводить с каждым билдом и количество этих тестов будет постоянно расти. Суть данного тестирования проверить, что ранее работавшая функциональность работает корректно. Выполнение таких тестов вручную рутинно и занимает большое количество времени. Так автоматизированное тестирование поможет решить эту проблему и на такие тесты будет уходить в разы меньше времени.	Планирование и разработка тест-кейсов.
Интеграционное тестирование.	Тесты, которые выполняются всего

<p>Данное тестирование используется для проверки взаимодействия компонент в продукте. Данные тесты проводятся постоянно и с каждым новым билдом. Поэтому автоматизируя такие тесты можно освобождается большое количество времени.</p>	<p>несколько раз. Внедряя автоматизацию, затраты могут не окупиться.</p>
<p>Модульное тестирование. Данное тестирование нацелено на проверку отдельных участков кода, которые проверяется на работоспособность, при этом нет зависимости от других частей приложения.</p>	<p>Тестирование удобства использования. Данный вид тестирования необходимо выполнять человеку так, как только он может оценить, насколько удобно пользоваться тем или иным продуктом.</p>
<p>Тестирование производительности и нагрузочное тестирование. Данный вид тестирования не может выполнять человек. Так как необходимо создавать высокую нагрузку на систему, например, тысячи пользователей, которые могут работать с приложением под различными условиями.</p>	<p>Стадии ранней разработки. Это ситуации, когда только разрабатывается функциональность и в нее вносятся большое количество изменений. Автоматизация долгий процесс, поэтому лучше тестировать вручную.</p>
<p>Дымовое тестирование. Данное тестирование направлено на проверку базового функционала приложения. Такие тесты просты в автоматизации, но очень важны для проекта. Они также проходят с</p>	<p>Низкая стабильность требований. Когда требования нестабильны и постоянно меняются, приходится постоянно менять код, что приводит к постоянному изменению тестов.</p>

каждым билдом.	
<p>Использование комбинаторных техник тестирования.</p> <p>В такой технике необходимо генерировать большое количество комбинаций и многократно выполнять тесты с этими значениями.</p>	<p>Тесты без понятных результатов. Для автоматизации обязательно должен быть ожидаемы результат, иначе нельзя доказать работоспособность функции.</p>
<p>Проверка внутренней функциональности веб-приложений.</p> <p>Под внутренней функциональностью понимаются ссылки на странице, доступность и открытие самих страниц. Такие действия достаточно рутинны, особенно если на страницы большое количество ссылок и необходимо проверить, что все они открываются.</p>	

Необходимо понимать, что автоматизация не всегда эффективна. Так как автоматизация требует времени и средств, то ее внедрение необходимо хорошо планировать, оценивая все риски.

## **2 Автоматизация тестирования с использованием Selenium**

В данной работе будет практика по написанию тест-кейсов для тестирования пользовательского интерфейса, которые представляют вершину в пирамиде автоматизации тестирования. Такие тесты проверяют полную работу системы и имитируют действия пользователя.

Существует большое количество инструментов для автоматизации тестирования как с открытым исходным кодом, так и коммерческих. Одним из наиболее популярных инструментов для автоматизации тестирования сайтов и веб-приложений является Selenium.

### **2.1 Используемые инструменты и технологии**

Selenium – это инструмент с открытым исходным кодом, который был разработан достаточно давно, однако в последние годы он активно развивается. Данная платформа поддерживает разные операционные системы и большое количество популярных браузеров, а также скрипты можно писать разных языках программирования, например, Python, C#, PHP, Java и другие.

Данный инструмент имеет свои преимущества и недостатки. Среди преимуществ можно отметить то, что Selenium обладает гибкостью и возможностью написания сложных скриптов для тестирования приложений. А одним из недостатков является то, что специалист по тестированию должен иметь высокую техническую квалификацию, то есть обладать знаниями в разработке программного обеспечения, а также быть готовым к затратам по времени на написание специальных библиотек и фреймов, которые обеспечивают выполнение необходимых функций в процессе автоматизированного тестирования.

У Selenium существует несколько продуктов:

- Selenium IDE – это дополнение к браузеру Firefox, которое используется для записи, редактирования и отладки тестов. Selenium IDE упрощает автоматизацию тестирования приложений и веб-программ.

- Selenium WebDriver содержит набор библиотек для различных языков программирования, которые позволяют управлять браузером из программы, написанной на данном языке программирования.
- Selenium Server принимает команды удаленно и выполняет их в браузере.
- Selenium Grid – это сеть из нескольких серверов Selenium, масштабирующая процесс автоматизации.

Для написания скриптов для UI-тестов будет использоваться Selenium WebDriver, который представляет из себя универсальный интерфейс для работы с разными браузерами напрямую из кода на языке программирования. Его универсальность заключается в том, что тесты, написанные один раз, можно запускать для разных браузеров. В качестве языка программирования был выбран Python.

### 2.1.1 Библиотека для написания автоматизированных тест-кейсов

Python содержит специализированную библиотеку для написания автоматизированных тест-кейсов PyTest. Преимущества и недостатки использования данной библиотеки представлены в таблице 3.

Таблица 3 – Преимущества и недостатки PyTest

Преимущества	Недостатки
<ul style="list-style-type: none"> <li>• Подробный отчёт с поддержкой цветовых схем из коробки.</li> <li>• PyTest не требует написания дополнительных специфических конструкций в тестах.</li> <li>• Для проверок используется стандартный assert из Python.</li> <li>• Возможность создания динамических фикстур</li> </ul>	<ul style="list-style-type: none"> <li>• PyTest требуется устанавливать дополнительно, так как он не входит в стандартный пакет библиотек Python.</li> <li>• Использование PyTest требует более глубокого понимания языка Python, чтобы разобраться, как применять фикстуры, параметризацию и другие возможности PyTest.</li> </ul>

(специальных функций, которые настраивают тестовые окружения и готовят тестовые данные).

- Дополнительные возможности по настройке фикстур.
- Параметризация тестов — для одного теста можно задать разные параметры (тест запустится несколько раз с разными тестовыми данными).
- Наличие маркировок (`marks`), которые позволяют маркировать тесты для их выборочного запуска.
- Возможность передавать дополнительные параметры через командную строку для настройки тестовых окружений.
- Большое количество плагинов, которые расширяют возможности PyTest и позволяют решать узкоспециализированные проблемы, что может сэкономить много времени.

Одним из важных моментов в использовании библиотеки PyTest является запуск тестов, так как он происходит по определенным правилам, которым необходимо следовать при создании скриптов для тест-кейсов:

- если мы не передали никакого аргумента в команду, а написали просто `pytest`, тест-раннер начнёт поиск в текущей директории;

- как аргумент можно передать файл, путь к директории или любую комбинацию директорий и файлов;
- дальше происходит рекурсивный поиск: то есть PyTest обойдет все вложенные директории;
- во всех директориях PyTest ищет файлы, которые удовлетворяют правилу `test_*.py` или `*_test.py` (то есть начинаются на `test_` или заканчиваются `_test` и имеют расширение `.py`)
- внутри всех этих файлов находит тестовые функции по следующему правилу:
  - все тесты, название которых начинается с `test`, которые находятся вне классов;
  - все тесты, название которых начинается с `test` внутри классов, имя которых начинается с `Test`.

### 2.1.2 Паттерн программирования для автоматизации

Page Object — это паттерн программирования, который очень популярен в автоматизации тестирования и является одним из стандартов при автоматизации тестирования веб-продуктов. Это также один из удобных способов структурировать свой код таким образом, чтобы его было удобно поддерживать, менять и работать с ним.

Основная идея состоит в том, что каждую страницу веб-приложения можно описать в виде объекта класса. Способы взаимодействия пользователя со страницей можно описать с помощью методов класса. В идеале тест, который будет использовать Page Object, должен описывать бизнес-логику тестового сценария и скрывать Selenium-методы взаимодействия браузером со страницей. При изменениях в верстке страницы не придется исправлять тесты, связанные с этой страницей. Вместо этого нужно будет поправить только класс, описывающий данную страницу.

Здесь применяются те же принципы, что и в разработке: мы хотим повысить читаемость кода и вынести в абстрактные методы все детали. Тесты

должны быть просто и понятно написаны, а повторяющиеся куски кода выделены в отдельные функции. В Page Object мы отделяем логику действий, например, авторизовать пользователя, от конкретной реализации (найти поле почты, ввести туда данные, найти поле пароля, ввести туда данные, найти кнопку и т.д.).

## 2.2 Процесс автоматизации тестовых сценариев

Процесс проведения тестирования с учетом внедренной автоматизации представлен на рисунке 2 в нотации BPMN.

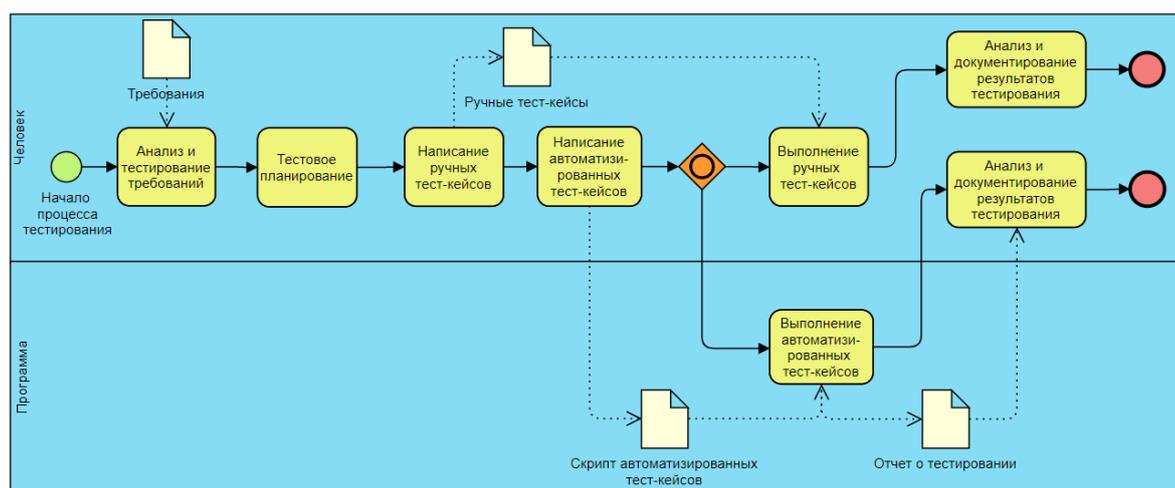


Рисунок 2 – Процесс проведения тестирования в нотации BPMN

Данный рисунок описывает участие человека и программы в процессе тестирования. На рисунке видно, что большую часть процессов выполняет человек, а программа для автоматизации занимается исключительно выполнением тест-кейсов. Но даже такой, казалось бы, маленький вклад автоматизации в процесс тестирования может быть ощутимо эффективен, с точки зрения скорости и освобождения ресурсов специалиста по тестированию.

### 2.2.1 Разработка тест-кейсов для ручного тестирования

В рамках данной работы были выбраны две основные функциональности, которые встречаются во всех разрабатываемых веб-приложениях:

1. Модуль авторизации;
2. Форма для добавления элемента.

Для каждого из этих модулей были разработаны тест-кейсы для проведения ручного тестирования.

Тест-кейсы или тестовые случаи – это артефакты тестирования, которые описывают совокупность шагов и входных данных для проверки реализации тестируемой функции или её части.

Тест-кейсы разделяются по ожидаемому результату на позитивные и негативные. Позитивный тест-кейс использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию. Негативный тест-кейс оперирует как корректными, так и некорректными данными (минимум 1 некорректный параметр) и ставит целью проверку исключительных ситуаций.

Тест-кейс имеет свою структуру:

1. ID. Могут разделяться в зависимости от тестового набора;
2. Название тест-кейса. Должно быть понятным и отражать суть выполняемой проверки;
3. Предусловие. Список действий, которые приводят систему к состоянию для проведения основной проверки;
4. Шаги для воспроизведения. Список действий, переводящих систему из одного состояния в другое;
5. Ожидаемый результат. Результат, который ожидается при воспроизведении шага;
6. Постусловие. Список действий, переводящих систему в первоначальное состояние. Не является обязательной частью.

Зачастую тест-кейсы группируются в тестовые наборы, которые относятся к одному тестируемому модулю. Это можно выделять также с помощью ID.

Тест-кейс не должен содержать зависимость от других тест-кейсов, нечеткие формулировки шагов и ожидаемого результата, отсутствие необходимой информации о прохождении тестов, а также излишней детализации.

Пример разработанных тест-кейсов для ручного тестирования представлен на рисунке 3.

ID	Название тест-кейса	Предусловие	Шаги по воспроизведению	Ожидаемый результат	Тестовые данные
1.1	Авторизация с корректными данными	Переход на страницу авторизации	1. В поле «Логин» ввести корректные данные. 2. В поле «Пароль» ввести корректные данные 3. Нажать на кнопку "Войти".	1. В поле «Логин» отображаются введенные данные. 2. В поле «Пароль» отображаются введенные данные в зашифрованном виде (в виде точек). 3. Вход выполнен. Осуществлен переход в профиль пользователя.	Логин: admin  Пароль: qweQWE123!
1.2	Авторизация с неверным значением поля "Логин"	Переход на страницу авторизации	1. В поле "Логин" ввести некорректные данные. 2. В поле "Пароль" ввести корректные данные. 3. Нажать на кнопку "Войти".	1. В поле "Логин" отображаются введенные данные. 2. В поле "Пароль" отображаются введенные данные в зашифрованном виде (в виде точек). 3. Авторизация не осуществлена. Появляется сообщение об ошибке.	Логин: admin  Пароль: qweQWE123!
1.3	Авторизация с неверным значением поля "Пароль"	Переход на страницу авторизации	1. В поле "Логин" ввести корректные данные. 2. В поле "Пароль" ввести некорректные данные. 3. Нажать на кнопку "Войти".	1. В поле "Логин" отображаются введенные данные. 2. В поле "Пароль" отображаются введенные данные в зашифрованном виде (в виде точек). 3. Авторизация не осуществлена. Появляется сообщение об ошибке.	Логин: admin  Пароль: 123qweQWE!

Рисунок 3 – Разработанные тест-кейсы для ручного тестирования

### 2.2.2 Разработка автоматизированных тест-кейсов

Для разработки автоматизированных тест-кейсов использовались инструменты, о которых было сказано в пункте 2.1. Разработка автоматизированных тест-кейсов для тестирования пользовательского интерфейса проводится после того, как разработана HTML-страница веб-приложения. Так как основа автоматизированных тест-кейсов для пользовательского интерфейса строится на применении CSS-селекторов или языка запросов XPath.

Итак, по разработанным тест-кейсам, пример которых представлен на рисунке 3, будут строиться автоматизированные тест-кейсы.

Структура автоматизированных тест-кейсов строится на паттерне программирования – Page Object, поэтому взаимодействия браузера с каждой страницей веб-приложения описывается в отдельном классе, так же, как и набор локаторов для каждой страницы. Локаторы содержат в себе константы, которые описывают доступ к элементам. Вынос этих констант упрощает

дальнейшую поддержку тестов. Например, если изменится доступ к элементу, то в коде его придется изменить всего один раз. Так диаграмма классов программы представлена на рисунке 4.

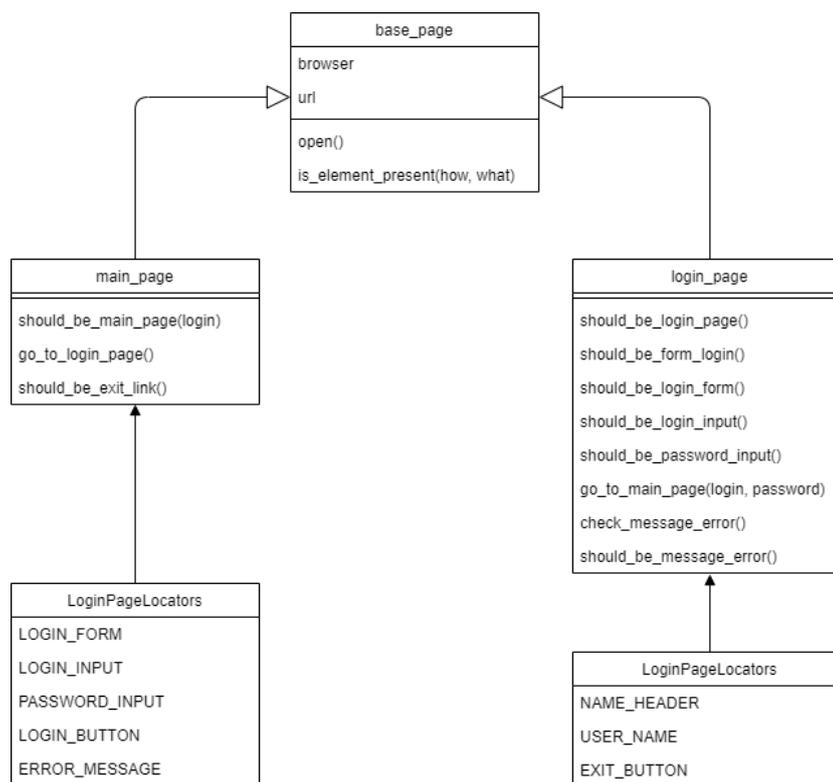


Рисунок 4 – Диаграмма классов

Помимо файлов, описывающих взаимодействия с каждой страницей и локаторов, так же существует конфигурационный файл.

Конфигурационный файл «conftest.py» предназначен для хранения фикстур или глобальных настроек. Фикстуры представляют из себя вспомогательные функции, которые не являются частью тестовых сценариев. В данном примере используется фикстура для открытия и закрытия браузера. Помимо фикстуры также используется и параметризация, которая позволяет настраивать тестовое окружение. В данном случае это выбор языка страницы. Делается это с помощью специальной функции «pytest\_addoption» и фикстуры «request». Так при запуске тестов можно выбирать язык, а если язык не выбран, то он будет установлен по умолчанию. Файл конфигурации представлен на рисунке 5.

```

import pytest
from selenium import webdriver
from selenium.webdriver.chrome.options import Options

@pytest.fixture(scope="function")
def browser(request):
    language = request.config.getoption("--language")
    print("\nstart browser for test..")
    options = Options()
    options.add_experimental_option('prefs', {'intl.accept_languages': language})
    browser = webdriver.Chrome(options=options)
    yield browser
    print("\nquit browser..")
    browser.quit()

def pytest_addoption(parser):
    parser.addoption('--language', action='store', default="ru",
                    help="Choose language: en or ru")

@pytest.fixture
def language(request):
    return request.config.getoption("--language")

```

Рисунок 5 – Файл конфигурации

Также на примере модуля авторизации рассмотрим файл, содержащий полученные автоматизированные тест-кейсы.

Файл с готовыми тест-кейсами для модуля авторизации представлен на рисунке 6. Данный файл описывает исключительно последовательность шагов для воспроизведения, благодаря вынесению взаимодействия браузера со страницей в отдельные классы. Помимо этого, для каждого теста используются маркировки и параметры. Маркировки предназначены отнесения тест-кейсов в отдельные группы, что помогает при запуске тестов. А параметры представляют входные данные для выполнения тест-кейсов, например, значение логина и пароля.

Общий алгоритм проверки модуля авторизации представлен на рисунке 7. Данный алгоритм описывает действия всех полученных автоматизированных тест-кейсов.

```

import pytest

from .pages.new_login_page import NewLoginPage
from .pages.main_page import MainPage

@pytest.mark.login
def test_guest_should_be_login_page(browser):
    link = "https://demoqa.com/login"
    login_page = NewLoginPage(browser, link)
    login_page.open()
    login_page.should_be_login_page()

@pytest.mark.login
@pytest.mark.parametrize('login', ["admin"])
@pytest.mark.parametrize('password', ["qweQWE123!"])
def test_guest_go_to_main_page_valid(browser, login, password):
    link = "https://demoqa.com/login"
    login_page = NewLoginPage(browser, link)
    login_page.open()
    login_page.should_be_login_page()
    login_page.go_to_main_page(login, password)
    main_page = MainPage(browser, browser.current_url)
    main_page.should_be_main_page(login)

```

Рисунок 6 – Файл с готовыми тест-кейсами для модуля авторизации

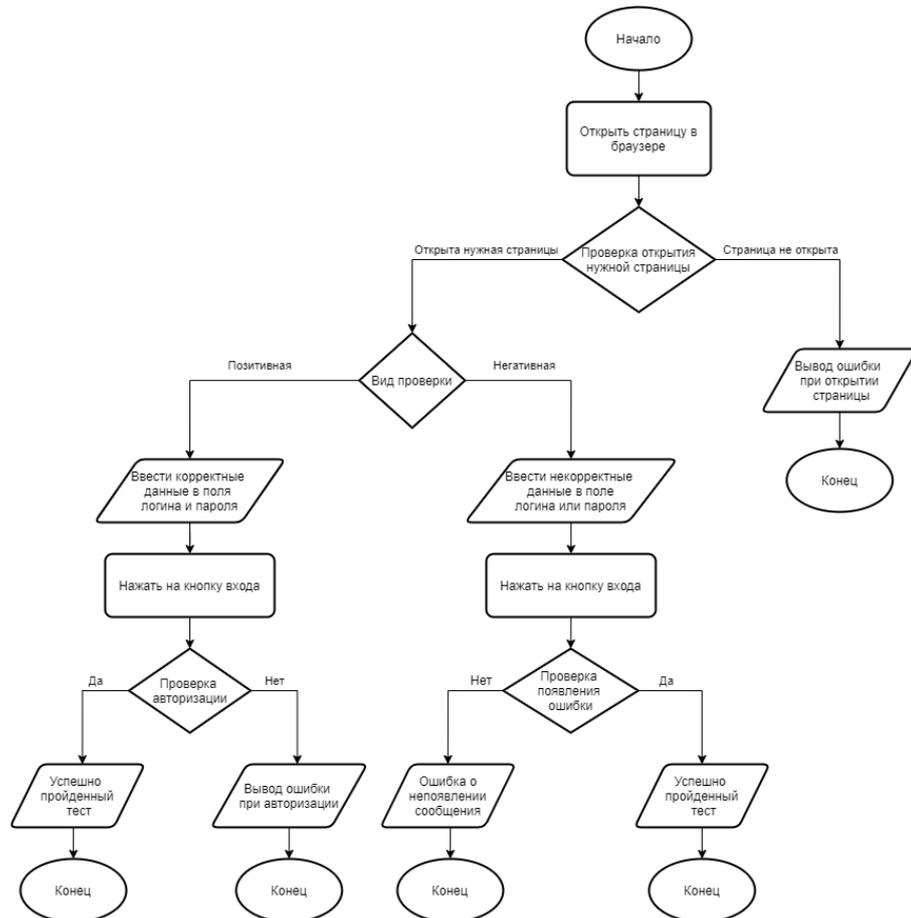
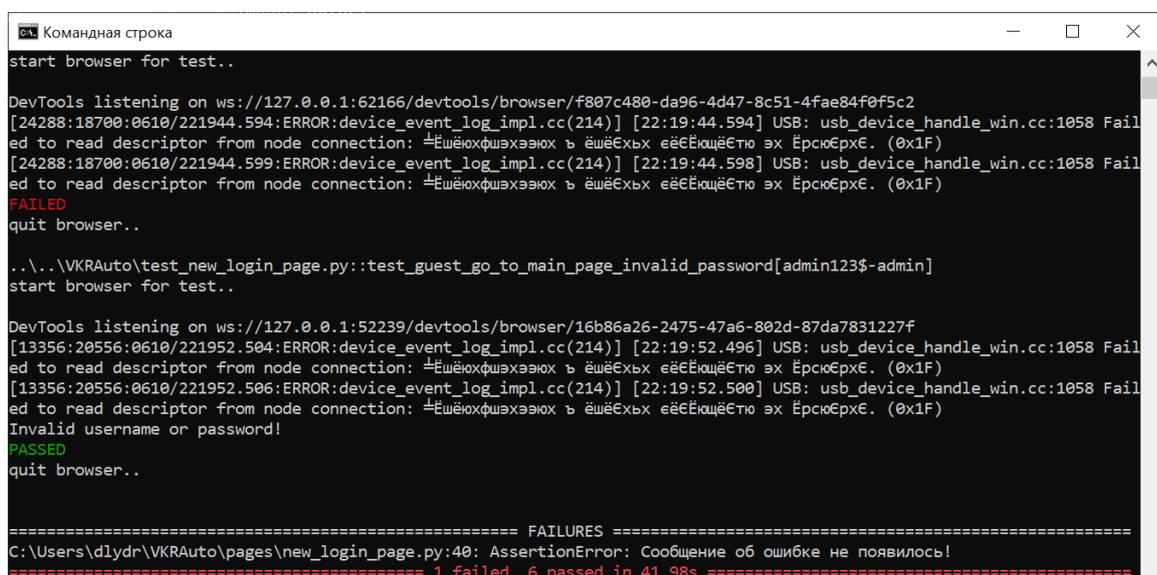


Рисунок 7 – Схема алгоритма для тестирования модуля авторизации

### 2.2.3 Выполнение автоматизированных тест-кейсов

Запуск автоматизированных с использованием библиотеки PyTest тест-кейсов осуществляется через консоль. У команды запуска выполнения автоматизированных тест-кейсов имеются различные параметры, которые позволяют выводить отчеты о тестировании в удобном виде. А также существуют параметры, которые запускают определенный набор тестовых сценариев. Тест-кейсы можно организовать в наборы с помощью маркировок. Отчет о тестировании выводимый в консоли необходимо анализировать и документировать, чем и занимается специалист по тестированию.

Результат выводимый в консоль представлен на рисунке 8. Там также отображается время, за которое выполнялось тестирование системы.



```
Командная строка
start browser for test..
DevTools listening on ws://127.0.0.1:62166/devtools/browser/f807c480-da96-4d47-8c51-4fae84f0f5c2
[24288:18700:0610/221944.594:ERROR:device_event_log_impl.cc(214)] [22:19:44.594] USB: usb_device_handle_win.cc:1058 Failed to read descriptor from node connection: {EШёюхфшээюх Ъ ёшёёхьх еёёёющёётю эх ёрсюёрхё. (0x1F)}
[24288:18700:0610/221944.599:ERROR:device_event_log_impl.cc(214)] [22:19:44.598] USB: usb_device_handle_win.cc:1058 Failed to read descriptor from node connection: {EШёюхфшээюх Ъ ёшёёхьх еёёёющёётю эх ёрсюёрхё. (0x1F)}
FAILED
quit browser..
..\VKRAuto\test_new_login_page.py::test_guest_go_to_main_page_invalid_password[admin123$-admin]
start browser for test..
DevTools listening on ws://127.0.0.1:52239/devtools/browser/16b86a26-2475-47a6-802d-87da7831227f
[13356:20556:0610/221952.504:ERROR:device_event_log_impl.cc(214)] [22:19:52.496] USB: usb_device_handle_win.cc:1058 Failed to read descriptor from node connection: {EШёюхфшээюх Ъ ёшёёхьх еёёёющёётю эх ёрсюёрхё. (0x1F)}
[13356:20556:0610/221952.506:ERROR:device_event_log_impl.cc(214)] [22:19:52.500] USB: usb_device_handle_win.cc:1058 Failed to read descriptor from node connection: {EШёюхфшээюх Ъ ёшёёхьх еёёёющёётю эх ёрсюёрхё. (0x1F)}
Invalid username or password!
PASSED
quit browser..

===== FAILURES =====
C:\Users\dlydr\VKRAuto\pages\new_login_page.py:40: AssertionError: Сообщение об ошибке не появилось!
===== 1 failed, 6 passed in 41.98s =====
```

Рисунок 8 – Результат автоматизированного тестирования для модуля авторизации

По результатам тестирования видно, что один тест-кейс упал с ошибкой, а 6 тест-кейсов были пройдены успешно. Также вывелось сообщение «Сообщение об ошибке не появилось», которое позволяет понять суть ошибки и быстрее определить ее локализацию.

### 3 Автоматизация тестирования с помощью RPA

Robotic Process Automation (RPA) – технология автоматизации бизнес-процессов, с использованием программных роботов и искусственного интеллекта. В свою очередь программный робот представляет из себя программу, которая имитирует действия человека, взаимодействуя с интерфейсом информационной системы, например, для сбора информации или манипулирования приложениями.

Программный робот имеет свое виртуальное рабочее пространство, в котором использует клавиатуру и мышь, для внесения данных и перемещения по экранным формам. Роботы способны выполнять работу не только на программном уровне, но и с использованием графического интерфейса, что является отличительной способностью технологии RPA. С использованием графического интерфейса робот сначала запоминает те или иные действия, а потом выполняет их сколько угодно раз.

Таким образом, внедрение RPA предоставляет пользователям инструмент для решения рутинных задач, который позволяет существенно сократить время работы бизнес-процесса. Это снижает загрузку сотрудников и повышает эффективность выполнения бизнес-задач, практически исключая возможность появления ошибок.

RPA роботы могут решать множество задач в различных сферах деятельности, таких как логистика, бухгалтерский учет и финансы, IT, клиентские сервисы и т.д. Для решения этих задач можно выделить основные навыки робота:

- Работа с данными. Роботы хорошо показывают себя в задачах, где нужно работать с разными информационными системами, например, переносить информацию из одного места в другое, разбивать и разносить информацию в разные места и т.д.;
- Принимать простые решения. Робот способен принимать простые решения, которые поддаются алгоритмизации. И в случае, если

возникает исключение, робот может обращаться за помощью к сотруднику;

- Проверка и валидация данных. Робот может сверять информацию из различных источников между собой, что позволяет эффективно проводить сверку информации на соответствия;
- Работа с неструктурированными данными. Роботы могут осуществлять поиск определенной информации в документах или договорах, а также проверять их с юридической точки зрения;
- Управление другими роботами и сотрудниками. Если роботов становится много, то существует возможность внедрения еще одного робота, который способен контролировать работу остальных роботов и составлять расписание их работы. Также робот способен сигнализировать сотруднику о наступлении какого-либо события;
- Обучение. Заранее запрограммированный робот может обучать новых сотрудников, не задействуя при этом ресурс человека.

В связи с навыками робота можно выделить следующие преимущества внедрения технологии RPA:

- Оптимизация производительности человеческих ресурсов. Один робот может заменить работу нескольких сотрудников;
- Повышение скорости автоматизируемого процесса;
- Исключение ошибок, которые могут возникать из-за потребностей человека. Точность выполнения задачи – 100%;
- Робот работает 24 часа в сутки, без отдыха и перерывов;
- При изменениях процесса, перестроить робота гораздо проще, чем человека;
- Робот может вести журналы своих действий.

### 3.1 UiPath Test Suite

Существует несколько платформ для разработки RPA роботов. Одна из этих платформ UiPath, которая является мировым лидером во всех аналитических рейтингах. Для автоматизации тестирования данная платформа выпустила специальный продукт UiPath Test Suite, его состав представлен на рисунке 9.

UiPath Test Suite построен для охвата всего процесса тестирования, от планирования и проектирования до реализации, выполнения и анализа результатов тестирования. Решение состоит из нескольких компонентов и каждый компонент отвечает конкретным потребностям: Test Manager предназначен для управления тестами, StudioPro - для их автоматизации, Orchestrator - для их распространения, а Robots - для выполнения тестов.

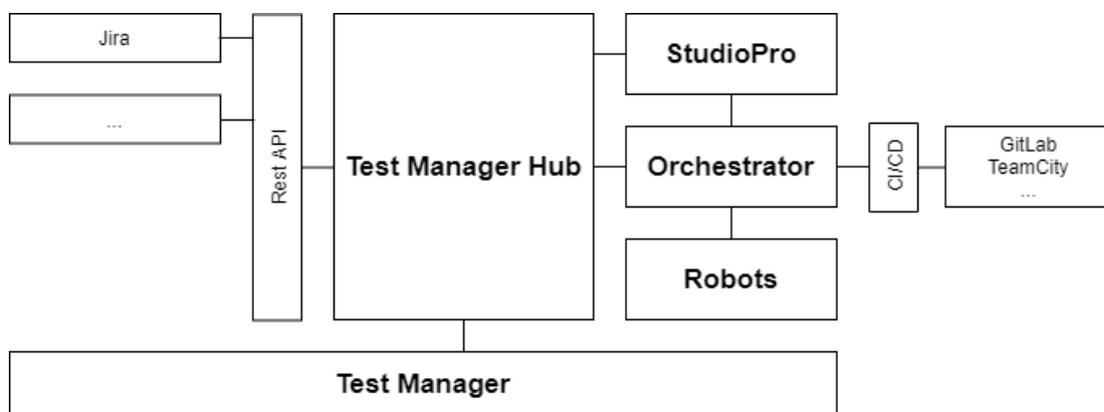


Рисунок 9 – Продукты UiPath Test Suite

Рассмотрим каждый компонент отдельно и спроецируем их предназначение на определенный ранее ЖЦ тестирования программного обеспечения.

Test Manager предназначен для определения требований, написания и выполнения ручных тестовых сценариев, привязку автоматизированных тестовых сценариев к требованиям, а также мониторинг и анализ результатов тестирования. Диспетчер дает возможность создать и определить требования в Test Manager, но в большинстве случаев требования фиксируются в других системах, например, Jira. Так существует Test Manager Hub, который

предназначен для импорта требований из внешних систем через определенные соединители.

Теперь перейдем к UiPath StudioPro, который представляет своеобразную IDE для создания автоматизированных тестовых сценариев и их локальной отладки. StudioPro позволяет запускать тестовые наборы локально по мере их создания. Однако есть возможность управлять тестовыми сценариями из одного места и запускать параллельно на нескольких машинах с помощью UiPath Orchestrator. Orchestrator – это веб-приложение, которое доступно как локально, так и в облаке. Тестовые сценарии попадают в Orchestrator после их публикации из StudioPro. Ход выполнения тестовых наборов можно отслеживать в Orchestrator. Также Orchestrator имеет разные способы запуска тестов: при каждой сборке проекта с помощью плагина Jenkins, по запланированному расписанию или вручную в любое время.

И последнее, но не менее важное: роботы - это агенты исполнения UiPath. Они подключены к Orchestrator, что позволяет запускать тестовые примеры с конкретными роботами или группами роботов, используя различные настройки машины.

### 3.1.1 UiPath Test Suite в жизненном цикле тестирования

Как уже было сказано выше решение UiPath Test Suite охватывает весь жизненный цикл тестирования программного обеспечения. Связь продуктов UiPath Test Suite и этапов жизненного цикла тестирования представлена на рисунке 10.

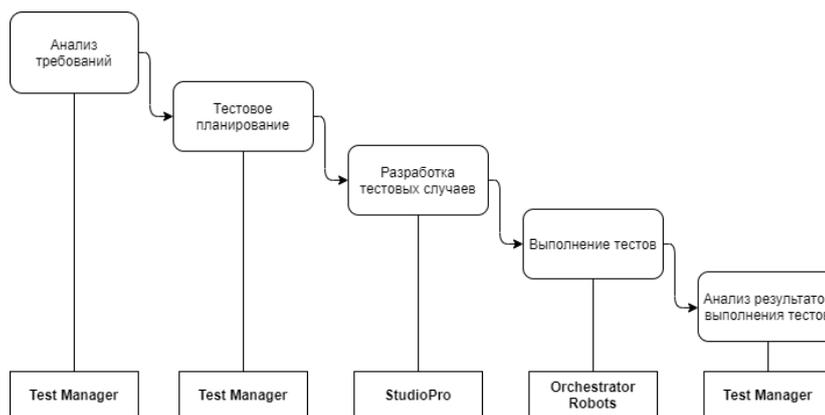


Рисунок 10 – Связь продуктов с жизненным циклом тестирования

Функция продуктов для каждого этапа:

- Анализ требований. Для данного этапа предназначен Test Manager, который позволяет создавать и управлять тестовыми проектами и соответствующими требованиями;
- Тестовое планирование. Для данного этапа также предназначен Test Manager, который позволяет записывать и отслеживать тестовые сценарии;
- Разработка тестовых случаев. Для данного этапа предназначена StudioPro, которая позволяет создавать автоматизированные тестовые сценарии, связывать тестовые сценарии с Test Manager и публиковать их в Orchestrator;
- Выполнение тестов. Для данного этапа предназначены Orchestrator и Robots, которые позволяют управлять опубликованными тестовыми сценариями, формировать тестовые наборы, выполнять тесты с помощью групп роботов, создавать и вести расписание выполнения тестов, а также управлять журналами выполнения тестов и результатами;
- Анализ результатов выполнения тестов. Для данного этапа также предназначен Test Manager, который позволяет следить за результатами выполнения тестов и анализировать их.

### **3.2 Создание тестовых сценариев в StudioPro**

Тестовые сценарии RPA в UiPath StudioPro строятся по формуле Given-When-Then.

Given – определяет контекст и предварительные условия тестового сценария, это могут быть: используемое приложение, входные данные или другая соответствующая информация.

When – содержит действия, которые будут выполнены.

Then – отображает ожидаемый результат.

Для реализации тестовых сценариев в StudioPro доступны три пакета, которые содержат все необходимые действия:

- UiPath.System.Activities – пакет системных действий, который содержит все основные действия, используемые для создания проектов автоматизации;
- UiPath.Testing.Activities – пакет действий по тестированию, который включает в себя действия, позволяющие легко проверять тестируемые системы;
- UiPath.UIAutomation.Activities – пакет действий UIAutomation, который содержит все основные действия, используемые для создания проектов автоматизации.

С помощью этих действий строятся тестовые сценарии в виде блок-схем.

### **3.2.1 Проверки**

На этапе Then в построении тестовых сценариев необходимо осуществить проверку по завершении тестового сценария. Для этого в UiPath StudioPro предусмотрены специальные действия. Сейчас их три вида:

- Verify Expression (Проверить выражение) – это действие проверяет выражение (например, равны ли две переменные), его результат является истинным или ложным;
- Verify Expression with Operator (Проверить выражение с помощью оператора) – это действие сравнивает результаты двух выражений, переменных или аргументов с использованием шести predefined операторов, его результат также является истинным или ложным;
- Verify Control Attribute (Проверить атрибут управления) – это наиболее универсальное действие, которое позволяет сравнивать свойство, возвращаемое другим действием, с выражением, переменной или аргументом. Его результат также может быть истинным или ложным.

### 3.2.2 Тестовые сценарии

Простые тестовые сценарии в StudioPro предполагают такие проверки, где нет различных вариаций во входных тестовых данных. Например, такими проверками можно считать переход по ссылкам, работоспособность или наличие кнопок и элементов на странице.

В данном случае простым тестовым сценарием можно считать предварительную авторизацию перед выполнением других действий в системе. Так в тестовом сценарии будет всего один набор входных данных – это корректные логин и пароль.

Чтобы построить оговоренный выше пример тестового сценария, необходимо воспользоваться набором действий, предоставляемых StudioPro. Изначально для любого тестового сценария строится рабочий процесс, который заключается в блок «Sequence». Этот процесс включает в себя последовательность шагов тестового сценария. Пример процесса для выполнения авторизации представлен на рисунке 11.

Рабочий процесс для тестового сценария представляет из себя

- открытие браузера с помощью действия «Open Browser». Используемое действие позволяет открывать браузер на странице с указанным URL, а также выбирать тип браузера. «Open Browser» поддерживаются такие браузеры, как IE, Firefox, Chrome, Edge и Custom;
- Блок «Open Browser» содержит большой блок «Do», который выполняет действия в данном браузере;
- Действие «Delay» устанавливает задержку прежде чем выполнять последующие действия;
- Действие «Type Into» позволяет вводить данные в элементы содержащие поля ввода. Выбор элементов для ввода считывается визуально со страницы браузера, при этом автоматически определяется нахождение элемента в иерархии элементов

страницы. Благодаря чему, можно в настройках изменять доступ к элементу;

- Действие «Click» инициирует нажатие на кнопку или ссылку. Существуют разные взаимодействия с мышью, например, двойной клик, одинарный клик или же наведение на элемент без клика.

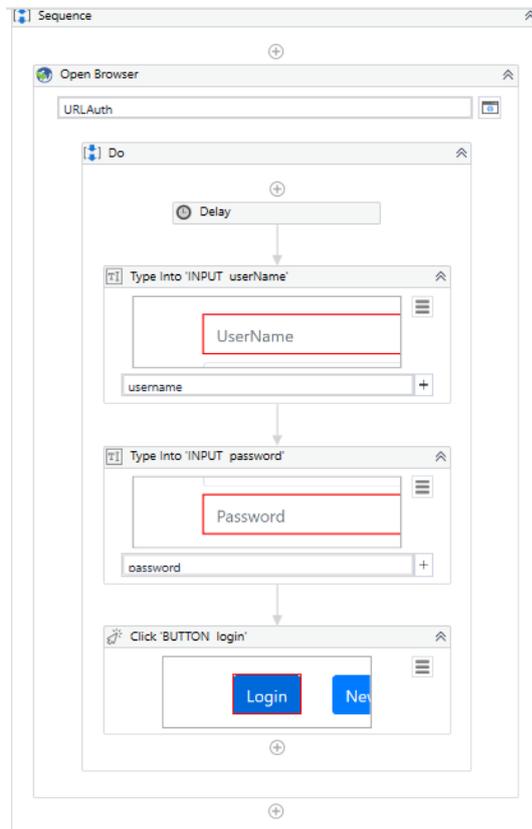


Рисунок 11 – Рабочий процесс тестового сценария

На основе созданного процесса формируются тестовые сценарии, представленные в форме Given-When-Then, которая представлена на рисунке 12.

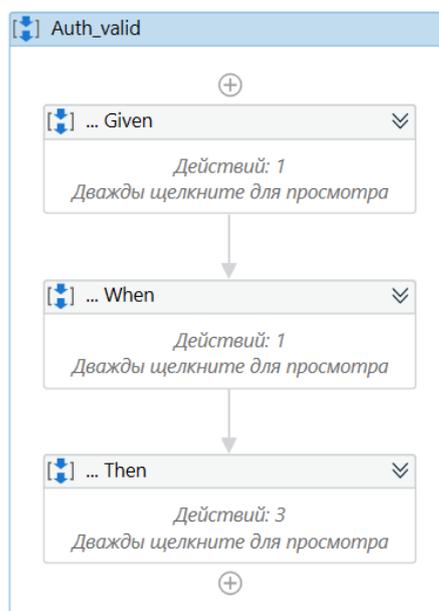


Рисунок 12 – Форма Given-When-Then

В этап When автоматически выставляется созданный ранее рабочий процесс, а этапы Given и Then формируются вручную. На этапе Given обычно инициализируются входные данные. Входные данные в простых тестовых сценариях представляют из себя аргументы, примерный список которых представлен на рисунке 13.

Имя	Направлен	Тип аргумента	Значение по умолчанию
username	Входной	String	"admin"
password	Входной	String	"qweQWE123!"
<i>Создать аргумент</i>			

Переменные | Аргументы | Импорт

Рисунок 13 – Список аргументов

Данный список определяет значение по умолчанию в случае, если оно не задано в блоке Given. Но обычно в блоке Given следует инициализировать эти данные. Это делается с помощью действия «Multiple Assign». Результат блока Given представлен на рисунке 14.

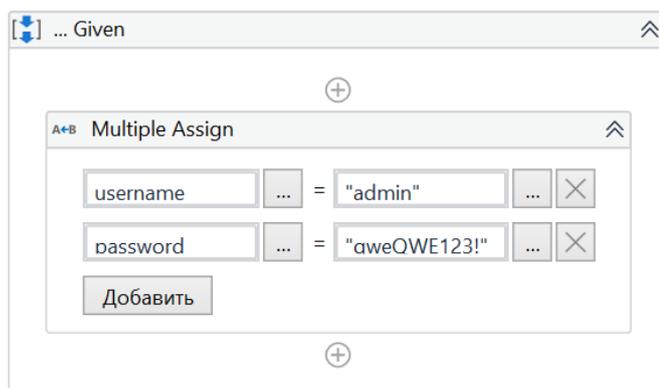


Рисунок 14 – Блок Given

Как уже было сказано выше, блок When автоматически содержит созданный ранее рабочий процесс, но в него необходимо передать аргументы, которые инициализируются на этапе Given. Готовый блок When представлен на рисунке 15.

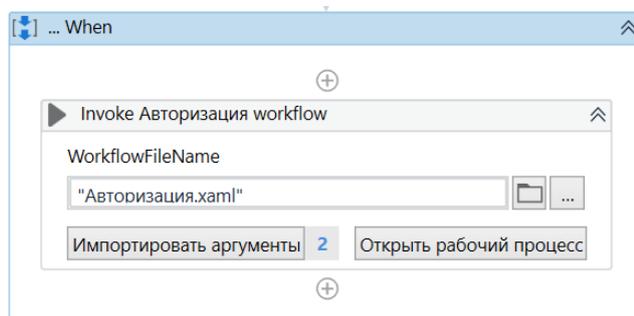


Рисунок 15 – Блок When

После выполнения основных шагов тестового сценария, необходимо проверить его успешность. Это проверяется в блоке Then. В пункте 3.2.1 были рассмотрены проверки, которые предоставляет StudioPro. В данном тестовом сценарии проверка осуществлялась с использованием действия «Verify Expression». Содержимое блока представлено на рисунке 16.

Здесь со страницы с помощью действия «Get Text» в переменную считывается текст определенного элемента. После чего с помощью действия «Verify Expression» происходит проверка переменной, в которую считался текст со страницы с переменной, которая была определена заранее и указывала на правильное поведение системы после успешной авторизации. И для того, чтобы успешно выполнять следующие действия необходимо выйти из профиля.

Для этого использовался блок «If», где при успешной авторизации осуществлялось нажатие на кнопку выхода с помощью действия «Click».

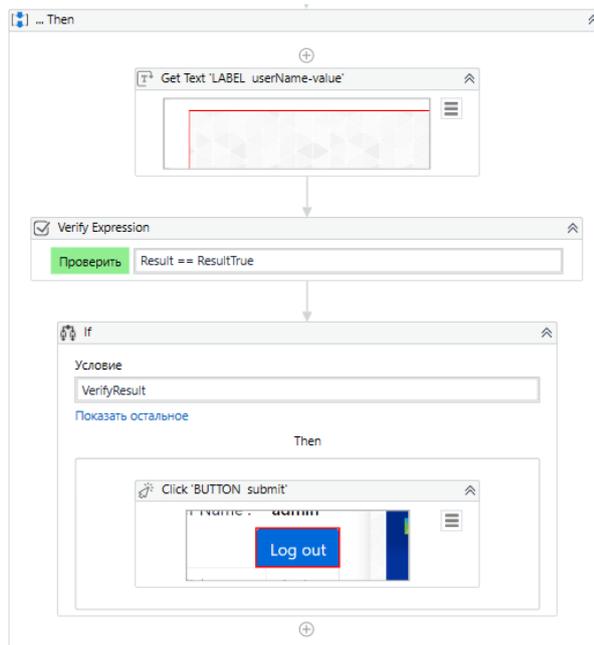


Рисунок 16 – Блок Then

В данном разделе были описаны действия для создания простого тестового сценария.

### 3.2.3 Тестовые сценарии на основе данных

В предыдущем разделе описывалось создание простого сценария, где в качестве входных данных использовался один набор. Но в большинстве случаев необходимо осуществлять такие проверки, где в качестве входных данных используется большое количество наборов этих данных для того, чтобы охватить большую часть вариантов использования системы. В таком случае автоматизация существенно позволяет снизить время выполнения таких тестов. В UiPath StudioPro также есть вариант создания тестов на основе данных.

При создании тестового сценария на основе данных необходимо загрузить набор входных значений. Обычно набор представляется в файле Excel, где первая строка обозначает название будущих атрибутов. Пример файла представлен на рисунке 17.

	A	B	C
1	<b>Username</b>	<b>Password</b>	<b>Message</b>
2	admni	qweQWE123!	Invalid username or password!
3	admni	123qweQWE\$	Invalid username or password!
4	admin	123qweQWE\$	Invalid username or password!
5	admin	QWEqwe123!	Invalid username or password!
6	admni	QWEqwe123!	Invalid username or password!
7			Invalid username or password!

Рисунок 17 – Набор входных данных

После также создается сценарий в форме Given-When-Then. Отличие заключается в том, что в блок Given нет необходимости передавать значения, так как они автоматически сохранились в системе в виде аргументов, которые передаются в блок When. В блоке Then также осуществляются проверки и в данных тестах использовалась проверка «Verify Expression with Operator», где два выражения проверялись с помощью predefined операторов. Блок-схему полученного тестового сценария можно увидеть на рисунке 18.

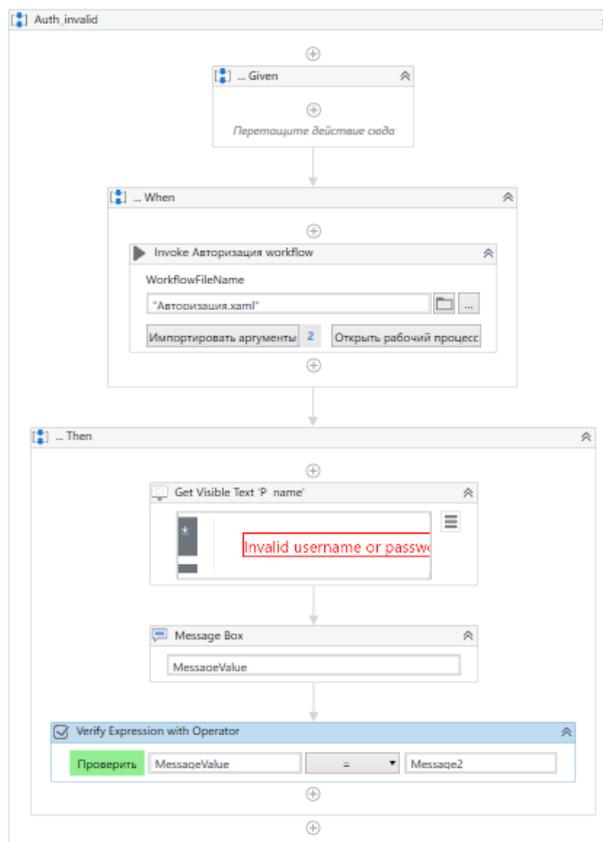


Рисунок 18 – Блок-схема тестового сценария

Запуск тестовых сценариев на основе данных можно осуществлять двумя способами:

- Запуск файла. Запускается тестовый сценарий с одним набором тестовых данных, который определен по умолчанию. Обычно это первый набор в файле Excel;
- Запуск файла с вариациями данных. Запуск тестового сценария запускается со всеми выбранными наборами входных данных. Перед запуском есть возможность выбрать необходимые наборы.

### 3.3 Выполнение тестовых сценариев

Запускать тестовые сценарии, разработанные в UiPath Test Studio можно различными способами. Первый и самый простой способ – это напрямую из StudioPro. Пример отчетности после запуска и проведения тестовых сценариев представлен на рисунке 19.

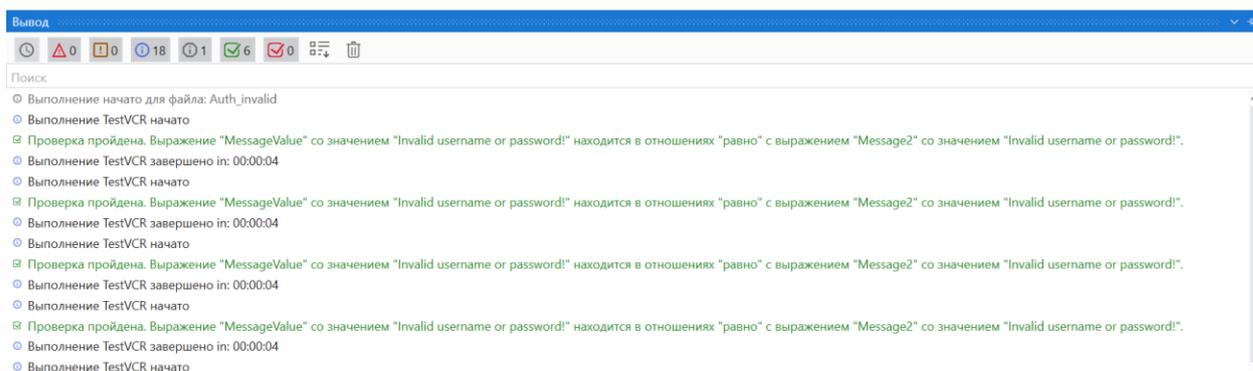


Рисунок 19 – Результат запуска автоматизированных тест-кейсов

Но также тесты можно запускать в специализированной среде, называемой Orchestrator, который управляет роботами, выполняющими тестовые сценарии. Процесс взаимодействия StudioPro с Orchestrator представлен на рисунке 20.

Из представленного процесса видно, что после публикации тест-кейсов из StudioPro в Orchestrator они становятся пакетами. Объединение пакета и среды робот представляет процесс. В свою очередь среда роботов – это набор роботов, который используется для развертывания процессов. Запуск тест-

кейсов производится специальными роботами, которые имеют тип «Testing». Также тест-кейсы в оркестраторы можно объединять тест-наборы, которые можно запускать самостоятельно, по расписанию или с использованием плагина CI/CD.

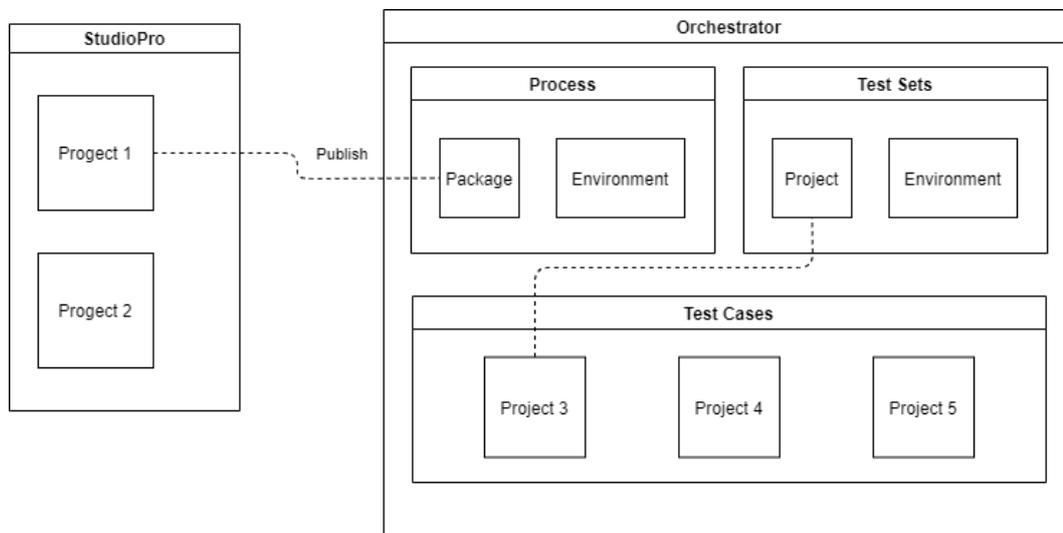


Рисунок 20 – Взаимодействие StudioPro и Orchestrator

Пример результатов запуска тест-кейсов из Orchestrator представлен на рисунке 21.

TEST RUN	VARIATION	RESULT	VERSION	ROBOT	MACHINE	STARTED	ENDED
Test Cases_Data-Driven Loan	1	Passed	1.0.41337707	R-Niclonut	NICIONUT	5 minutes ago	5 minutes ago
Test Cases_Data-Driven Loan	2	Passed	1.0.41337707	R-Niclonut	NICIONUT	5 minutes ago	4 minutes ago
Test Cases_Data-Driven Loan	3	Passed	1.0.41337707	R-Niclonut	NICIONUT	4 minutes ago	4 minutes ago
Test Cases_Data-Driven Loan	4	Passed	1.0.41337707	R-Niclonut	NICIONUT	4 minutes ago	4 minutes ago
Test Cases_Data-Driven Loan	5	Passed	1.0.41337707	R-Niclonut	NICIONUT	4 minutes ago	3 minutes ago
Test Cases_Data-Driven Loan	6	Passed	1.0.41337707	R-Niclonut	NICIONUT	3 minutes ago	3 minutes ago
Test Cases_Data-Driven Loan	7	Passed	1.0.41337707	R-Niclonut	NICIONUT	3 minutes ago	3 minutes ago
Test Cases_High Volume Loan		Passed	1.0.41337707	R-Niclonut	NICIONUT	3 minutes ago	3 minutes ago

Рисунок 21 – Запуск тестов из Orchestrator

Для каждого результата можно посмотреть детали и подробную информацию о проверке и о том, как она была построена в StudioPro.

## 4 Анализ полученных результатов

### 4.1 Сравнение инструментов автоматизации тестирования

В результате проведения автоматизации тестирования с помощью двух инструментов был проведен их сравнительный анализ, который представлен в таблице 4.

Таблица 4 – Сравнительный анализ инструментов автоматизации тестирования

Характеристика	Автоматизация с помощью Selenium	Автоматизация с помощью RPA
Знание языков программирования	Необходимо знать хотя бы один язык программирования, поддерживаемый данным продуктом	Нет необходимости знать языки программирования
Тестовые сценарии	Строятся в виде скрипта	Строятся в виде блок схем
Доступ к элементам на странице	Осуществляется с помощью CSS-селекторов и языка запросов XPath	Осуществляется запись элемента на странице, доступ к элементу определяется системой
Вовлеченность в ЖЦ тестирования	Только на этапе разработки и выполнения тестовых сценариев	На всех этапах ЖЦ тестирования
Настроить запуск тестов по расписанию	Можно, но используя другие средства для планирования	Можно с помощью данной системы
Стоимость продукта	Бесплатно	Платно, необходимо получить лицензию

По результатам сравнительного анализа можно выделить одно большое преимущество использования технологии RPA и инструмента UiPath для автоматизации тестирования, это наличие единого автоматизированного пространства, которое охватывает все этапы ЖЦ тестирования. При использовании данного инструмента нет необходимости использовать другие системы и средства, например, для хранения ручных тестовых сценариев или для хранения требований. Помимо этого, данный продукт легко интегрируется с другими продуктами, что позволит быстрее вовлечь его в существующий рабочий процесс тестирования. Но стоит отметить, что у продукта также есть и большой недостаток – это его стоимость. Именно этот недостаток не позволит большому количеству компаний внедрить данный инструмент для автоматизации процесса тестирования.

#### 4.2 Сравнение результатов проведения тестирования

В качестве объектов для сравнения результатов проведения тестирования был взят процесс проведения ручного тестирования и автоматизированного тестирования с использованием Selenium и RPA от разработки тест-кейсов до их выполнения.

Основные характеристики и результаты по данным характеристикам представлены в таблице 5.

Таблица 5 – Сравнительный анализ проведения тестирования

Характеристика	Ручное тестирование	Автоматизированное тестирование с помощью Selenium	Автоматизированное тестирование с помощью RPA
Время разработки тест-кейсов, мин	21	53	32
Время выполнения тест-кейсов, мин	7	0,7	0,57

По результатам сравнительного анализа видно, что время на разработку автоматизированных тест-кейсов уходит больше, чем на разработку ручных тест-кейсов. При этом разработка тест-кейсов на RPA быстрее, чем на Selenium. Но время на разработку тест-кейсов компенсируется их выполнением, так как выполнение автоматизированных тест-кейсов проходит в разы быстрее, чем выполнение ручных тест-кейсов. И также выполнение с помощью RPA оказалось быстрее, чем с помощью Selenium. Но это также не означает, что автоматизировать нужно все модули и функции.

### 4.3 Оценка эффективности внедрения автоматизации

Для оценки эффективности от внедрения автоматизации был использован показатель эффективности, который учитывает не денежную эффективность, а эффективность, зависящую от времени работы.

В качестве показателя для расчёта эффективности был выбран ROI. ROI – это коэффициент возврата инвестиций.

Так как расчет производится с точки зрения использования временных ресурсов, то формула будет выглядеть следующим образом:  $ROI = \frac{TM - TA}{TM}$ , где  $TM$  – временные затраты на проведение ручного тестирования,  $TA$  – временные затраты на проведение автоматизированного тестирования.

При этом  $TM = TM_0 + \sum_{i=1}^n (TM_e + TM_a + TM_m)$ , где  $TM_0$  – это время, потраченное на разработку первоначальных тест-кейсов,  $TM_e$  – время, потраченное на однократное выполнение набора тест-кейсов,  $TM_a$  – предполагаемое время на анализ результатов,  $TM_m$  – время на сопровождение тест-кейсов.

А  $TA = TA_0 + \sum_{i=1}^n (TA_e + TA_a + TA_m)$ , где  $TA_0$  – это время, потраченное на разработку первоначальных тест-кейсов,  $TA_e$  – время, потраченное на однократное выполнение набора тест-кейсов,  $TA_a$  – предполагаемое время на анализ результатов,  $TA_m$  – время на сопровождение тест-кейсов.

При расчете данного показателя предположим, что  $TM_a = 5$  минут, а  $TM_m = 5$  минут.

Для автоматизированного тестирования с помощью Selenium, предположим  $TA_a = 5$  минут, а  $TA_m = 8$  минут.

Для автоматизированного тестирования с помощью RPA, предположим  $TA_a = 7$  минут, а  $TA_m = 7$  минут.

Данные предположения субъективны.

Расчет будет производиться для 15 выпусков продукта, результаты эффективности представлены в таблице 6.

Таблица 6 – Результаты эффективности

	1	2	3	4	5	6	7	8	9	10	11
Selenium	-0,8	-0,5	-0,3	-0,2	-0,14	-0,09	-0,06	-0,03	-0,01	0,01	0,02
RPA	-0,2	-0,1	-0,05	-0,01	0,01	0,03	0,04	0,05	0,06	0,07	0,075

По полученным результатам были построены диаграммы: диаграмма временных затрат, которая показывает время общее время, затраченное на тестирование и диаграмма эффективности, которая показывает эффективность от внедрения автоматизации. Данные диаграммы представлены на рисунке 22.

По результатам диаграмм можно сказать, что эффективность от автоматизации тестирования на Selenium возникает только после 10 выпуска, тогда как эффективность от автоматизации на RPA возникает уже после 5 выпуска. Из этого можно сделать вывод, что для рассматриваемого модуля эффективность от автоматизации на RPA в два раза выше, чем от автоматизации на Selenium.

Но также стоит отметить, что эффективность Selenium станет выше эффективности RPA к 23 выпуску. А это значит, что не всегда RPA будет эффективнее. Все зависит от планируемого количества выпусков в системе и от авторизуемого модуля.

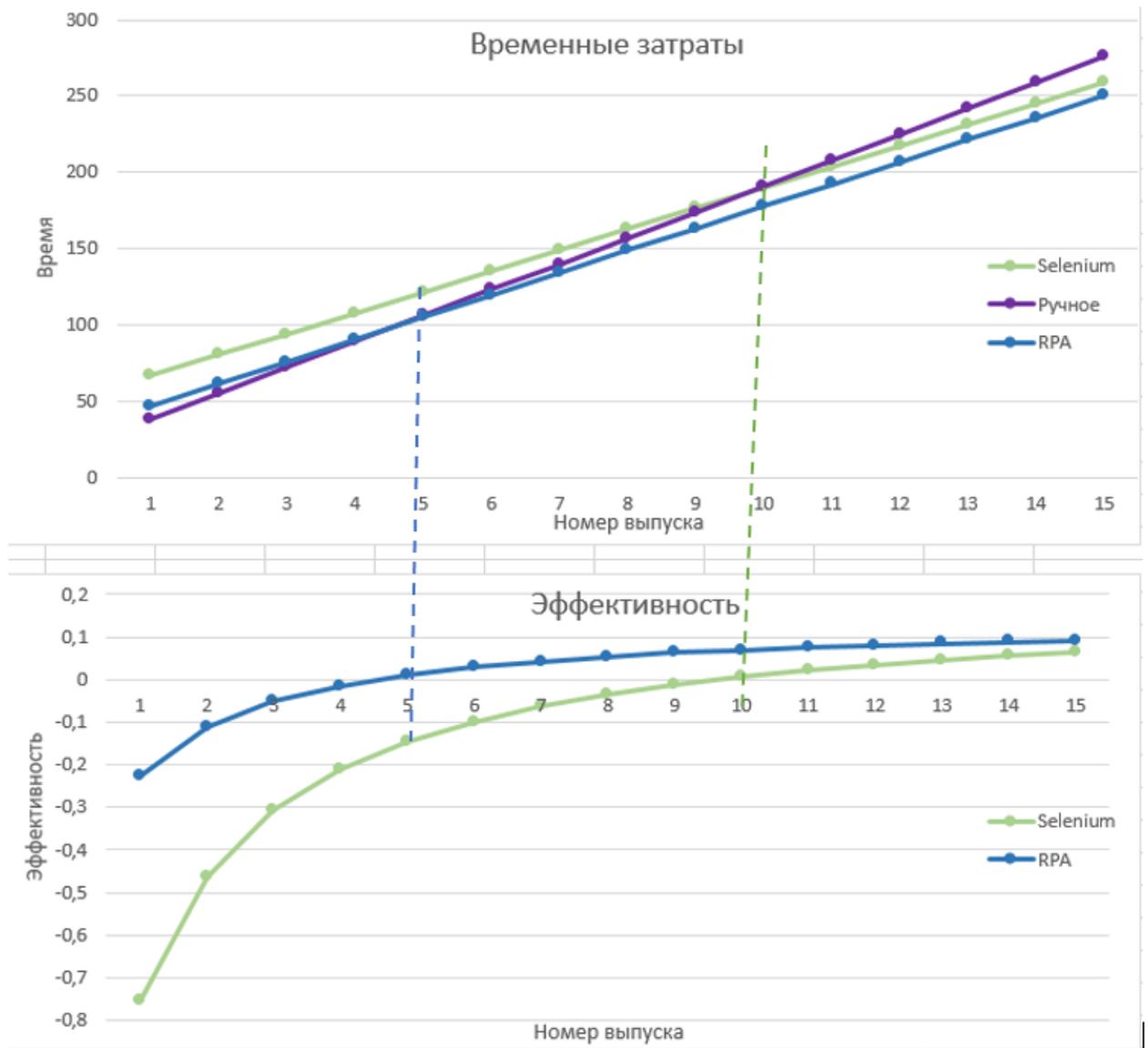


Рисунок 22 – Диаграммы эффективности

## **5 Социальная ответственность**

### **5.1 Введение**

Программные системы являются неотъемлемой частью повседневной жизни человека, так как они охватывают практически все отрасли. Поэтому очень важно создавать качественный продукт. За качество в цикле разработки программного обеспечения отвечает процесс тестирования. Так одной из задач данного проекта является автоматизация процесса тестирования. Автоматизация позволит частично исключить человека из процесса тестирования. Существует большое количество инструментов и технологий для автоматизации, но цель данного проекта применить относительно новую технологию RPA к процессу автоматизации тестирования.

Технологиями и инструментами автоматизации тестирования пользуются QA-инженеры по автоматизации. Данная работа будет проводиться в офисах различных компаний, где рабочие места оборудованы всей необходимой оргтехникой, например, монитор, системный блок, клавиатура, мышь, офисный стол, кресло и др.

Данный раздел направлен на определение правовых норм трудовой деятельности и мероприятий по организации рабочего места сотрудника. В связи с чем необходимо определить вредные и опасные факторы, влияющие на сотрудника в процессе его работы над автоматизацией тестирования. А также определить мероприятия по снижению воздействия этих факторов. Помимо человека процесс производства влияет на окружающую среду, что также будет отражено в данном разделе.

### **5.2 Правовые и организационные вопросы обеспечения безопасности**

Трудовой кодекс Российской Федерации нацелен на установление государственных гарантий трудовых прав и свобод граждан, создание благоприятных условий труда, защите прав и интересов работников и работодателей.

Согласно трудовому кодексу Российской Федерации нормальная продолжительность рабочего времени не может превышать 40 часов в неделю. При этом в течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов.

Компания обязана предоставлять работнику ежегодный основной оплачиваемый отпуск предоставляется работникам продолжительностью 28 календарных дней.

Минимальный размер оплаты труда устанавливается одновременно на всей территории Российской Федерации федеральным законом и не может быть ниже величины прожиточного минимума трудоспособного населения. А работа в выходной или нерабочий праздничный день оплачивается не менее чем в двойном размере.

В целях обеспечения прав и свобод человека и гражданина работодатель и его представители при обработке персональных данных работника обязаны соблюдать общие требования, описанные в статье 86 ТК РФ.

Организация рабочего места для специалиста по автоматизации тестирования представлена в «ГОСТ 12.2.032-78 ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования.». Согласно данному стандарту при проектировании оборудования и организации рабочего места следует учитывать антропометрические показатели женщин (если работают только женщины) и мужчин (если работают только мужчины), если оборудование обслуживают женщины и мужчины - общие средние показатели женщин и мужчин.

Конструкция производственного оборудования и рабочего места должна обеспечивать оптимальное положение работающего, что достигается регулированием высоты рабочей поверхности, сиденья и пространства для ног.

Взаимное расположение элементов рабочего места должно обеспечивать возможность осуществления всех необходимых движений и перемещений для эксплуатации и технического обслуживания оборудования. А также взаимное расположение элементов рабочего места должно способствовать оптимальному режиму труда и отдыха, снижению утомления оператора, предупреждению появления ошибочных действий.

### 5.3 Производственная безопасность

В качестве вредных и опасных факторов были выделены следующие: повышенный уровень шума, недостаточная освещенность рабочей зоны, повышенный уровень электромагнитных излучений и отклонения показателей микроклимата. Вышеперечисленные факторы описаны в таблице 7.

Таблица 7 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Этапы работ			Нормативные документы
	Разработка	Изготовление	Эксплуатация	
Повышенный уровень шума	+	+	+	ГОСТ 12.1.003-2014 ССБТ. Шум. Общие требования безопасности. СП 51.13330.2011. Защита от шума. Актуализированная редакция СНиП 23-03-2003.
Недостаточная освещенность рабочей зоны	+	+	+	СП 52.13330.2016 Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*.
Повышенный уровень	+	+	+	ГОСТ 12.1.006-84 ССБТ.

электромагнитных излучений				Электромагнитные поля радиочастот. Общие требования безопасности.
Отклонения показателей микроклимата	+	+	+	ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны.

### 5.3.1 Анализ опасных и вредных производственных факторов

#### 1. Повышенный уровень шума.

В качестве источника данного фактора могут быть кратковременный удары или взрывы, а также шум от машин или разговоров коллег.

Шум на рабочем месте оказывает раздражающее влияние на работника, повышает его утомляемость, а при выполнении задач, требующих внимания и сосредоточенности, способен привести к росту ошибок и увеличению продолжительности выполнения задания. Длительное воздействие шума влечет тугоухость работника вплоть до его полной глухоты.

Повышенный шум на рабочем месте оказывает вредное влияние на организм работника в целом, вызывая неблагоприятные изменения в его органах и системах. Длительное воздействие такого шума способно привести к развитию у работника потери слуха, увеличению риска артериальной гипертензии, болезней сердечно-сосудистой, нервной системы и др. При этом специфическим клиническим проявлением вредного действия шума является стойкое нарушение слуха (тугоухость), рассматриваемое как профессиональное заболевание.

Предельно допустимые и допустимые уровни звукового давления, дБ (эквивалентные уровни звукового давления, дБ), допустимые эквивалентные и максимальные уровни звука на рабочих местах отражены в таблице 8.

Таблица 8 – Предельно допустимые и допустимые уровни звукового давления, уровни звука

Назначение помещений	Уровни звукового давления дБ, в октавных полосах частот со среднегеометрическими частотами, Гц									Уровень звука $L_A$ , дБА	Максимальный уровень звука $L_{Амакс}$ , дБА
	31,5	63	125	250	500	1000	2000	4000	8000		
Помещения офисов, рабочие помещения	86	71	61	54	49	45	42	40	38	50	65

Обеспечение безопасности при воздействии шума на работника является комплексным мероприятием с участием разных сторон. Национальным законодательством устанавливаются требования к сторонам, являющимися прямыми или косвенными участниками трудового процесса, по обеспечению безопасности воздействия шума и меры, принимаемые к сторонам при несоблюдении установленных требований.

Работодатель должен обеспечить безопасности при воздействии шума на работников посредством принятия соответствующих мер, например, оценить риски, с учетом рисков проектировать рабочие места, составлению комплексных программ сохранения слуха работника. Работник, оповещенный работодателем о возможных рисках, связанных с воздействием шума, и о необходимости использовать в целях снижения рисков средства индивидуальной защиты от шума, должен следовать установленным работодателем правилам безопасного ведения работ и применения средств индивидуальной защиты от шума.

## 2. Недостаточная освещенность рабочей зоны.

Большую роль в создании благоприятных условий для работоспособности на предприятии играет освещение. Отсутствие хорошего освещения может привести к профессиональным заболеваниям, а также ухудшению концентрации работников. Работа специалиста по автоматизации тестирования в основном проводится за монитором персонального компьютера. В результате недостаточной освещенности рабочего места у работника может ухудшиться зрение, а также возникнуть переутомление. То же самое происходит и при избыточном освещении помещения.

Причинами недостаточной освещенности могут быть недостаточность естественного освещения или недостаточность искусственного освещения, пониженная контрастность. Таким образом, в рабочем помещении должны присутствовать естественное и искусственное освещение надлежащего уровня.

В таблице 9 подставлены нормы для освещенности рабочей зоны.

Таблица 9 – Нормы освещения рабочей зоны

Высокой точности	Характеристика зрительной работы	Искусственное освещение		Естественное освещение	Совмещенное освещение
		Освещенность, лк	Сочетание нормируемых величин объединенного показателя дискомфорта UGR и коэффициента пульсации		
От 0,30 до 0,50	Наименьший или эквивалентный размер объекта различения, мм				
Ш	Разряд зрительной работы				
б	Подразряд зрительной работы				
Малый Средний	Контраст объекта с фоном				
Средний Темный	Характеристика фона				
1000	Всего	при системе комбинированного освещения			
200	В том числе от общего				
400		при системе общего освещения			
25		UGR, не более			
15		K <sub>п</sub> , %, не более			
-		при верхнем или комбинированном освещении			
-		при боковом освещении			
3,0		при верхнем или комбинированном освещении			
1,2		при боковом освещении			

### 3. Повышенный уровень электромагнитных излучений

Источником электромагнитных излучений на рабочем месте специалиста по автоматизации тестирования может быть персональный компьютер.

Влияние электромагнитных полей на тело человека проявляется в ряде негативных последствий для организма в целом, а также отдельных его составляющих. В наибольшей степени страдает нервная система, а также сердечно-сосудистая. Сначала возникают такие признаки, как головная боль и головокружение, общая слабость, нарушение сна и т.д. Страдает давление, изменения также проявляются в увеличении или уменьшении артериального давления. Далее следует замедление пульса, боли в сердце (могут сопровождаться тахикардией или брадикардией), выпадение волос и ломкость ногтевых пластин. На ранних стадиях поражения последствиями электромагнитного излучения, носят обратимый характер. Есть возможность избавиться от таких последствий путем прекращения воздействия негативного фактора, а также после проведения симптоматической терапии.

Согласно стандарту «ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности» уровни ЭМП на рабочих местах контролируются измерением в диапазоне частот 60 кГц - 300 МГц напряженности электрической и магнитной составляющих, в диапазоне частот 300 МГц - 300 ГГц плотности потока энергии ЭМП с учетом времени пребывания персонала в зоне облучения.

#### 4. Отклонение показателей микроклимата.

Микроклимат в помещении – один из самых необходимых для обеспечения благоприятных условий труда работников производственный фактор, поскольку он оказывает большое влияние на тепловое самочувствие человека.

Согласно «ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны» к показателям, характеризующим микроклимат относятся: температура воздуха, относительная влажность воздуха, скорость движения воздуха и интенсивность теплового излучения.

Оптимальные показатели микроклимата распространяются на всю рабочую зону, допустимые показатели устанавливаются дифференцированно для постоянных и непостоянных рабочих мест. Оптимальные и допустимые

показатели температуры, относительной влажности и скорости движения воздуха в рабочей зоне производственных помещений должны соответствовать значениям, указанным в таблице 10.

Допустимые величины показателей микроклимата устанавливаются в случаях, когда по технологическим требованиям, техническим и экономическим причинам не обеспечиваются оптимальные нормы.

Таблица 10 – Оптимальные и допустимые нормы температуры, относительной влажности и скорости движения воздуха в рабочей зоне производственных помещений

Период года	Температура, °С		Относительная влажность, %		Скорость движения, м/с	
	Оптимальная	Допустимая	Оптимальная	Допустимая	Оптимальная	Допустимая
Холодный	22-24	21-25	40-60	75	0,1	Не более 0,1
Теплый	23-25	22-28	40-60	55	0,1	0,1-0,2

В целях профилактики неблагоприятного воздействия микроклимата используются следующие защитные мероприятия: естественная вентиляция (аэрация), системы кондиционирования воздуха, воздушное душирование рабочих мест, помещения для отдыха и обогрева (охлаждения), регламентация времени работы, время пребывания на рабочих местах при температуре воздуха выше или ниже допустимых величин регламентируется санитарными правилами.

### 5.3 Экологическая безопасность

В данном разделе рассматривается воздействие на окружающую среду в результате применения технологии RPA в автоматизации тестирования.

Процесс автоматизации тестирования с использованием данной технологии может нанести вред, как литосфере, так и гидросфере с атмосферой. Как уже упоминалось выше при автоматизации тестирования использует следующая оргтехника: монитор, системный блок, клавиатура и мышь и т.д. При выходе из строя данного оборудования или замены старого оборудования на новое, вышеперечисленная оргтехника списывается и отправляется на специальный склад, где принимается решение об ее утилизации.

Неправильный процесс утилизации может нанести вред литосфере, гидросфере и атмосфере выбросом вредных веществ после утилизации оборудования, например, если выбросить некоторые детали компьютера на свалку, то можно нанести природе непоправимый вред. Содержание в них олова, свинца, цинка и ртути приводит к уничтожению живых организмов.

#### **5.4 Безопасность в чрезвычайных ситуациях**

Чрезвычайные ситуация классифицируются по следующим признакам: природного характера, техногенного характера, биолого-социального характера и экологического характера.

Так к возможным чрезвычайным ситуациям на рабочем месте специалиста по автоматизации тестирования можно отнести: пожар, взрыв и обрушения зданий, аварии на электроэнергетических системах, относящиеся к техногенным чрезвычайным ситуациям, пандемия и эпидемия, относящиеся к биолого-социальным чрезвычайным ситуациям.

Самой типичной чрезвычайной ситуацией на рабочем месте специалиста по автоматизации тестирования может быть пожар.

Причиной пожара может стать:

- короткое замыкание, которое может случиться из-за изношенности оборудования, несоблюдения графика профилактических работ, неверного соединения провод или механического повреждения кабеля;

- перегрузка цепей питания из-за высокого напряжения, неполного короткого замыкания или неисправности эксплуатируемого оборудования;
- Нагрев металлических конструкций при попадании на них напряжения с оголенных кабелей;
- Контакт тепловых приборов с легковоспламеняющимися материалами и др.

Во избежание пожаров необходимо строго соблюдать правила – контроль исправности приборов, своевременный ремонт, выполнение требований пожарной безопасности. Также оснащение рабочих мест средствами первичного пожаротушения и своевременное обучение персонала действиям при обнаружении очага возгорания позволит снизить частоту аварийных ситуаций, связанных с воспламенением, уменьшить риск человеческим жизням и сократить материальный ущерб.

Если же все-таки пожар произошел необходимо:

- сообщить о пожаре по единому телефону пожарных и спасателей (101 - с мобильного, 01 - со стационарного);
- до прибытия профессионалов постараться сделать всё возможное для спасения других людей, ценного имущества и, конечно, себя;
- по возможности найти план эвакуации в случае пожара. На нём указаны все направления и пути возможной эвакуации, расположение лестниц, запасных выходов и телефонов;
- дышать через влажный носовой платок или рукав и двигаться к выходу, держась за стены и поручни.

## **5.5 Вывод по разделу**

В ходе написания раздела были рассмотрены правовые нормы трудового кодекса Российской Федерации, которые регламентируют взаимоотношения между работником и работодателем. Также рассмотрены мероприятия по

организации рабочих мест и рабочей зоны в целом. Были выделены наиболее вредные факторы, влияющие на сотрудника при выполнении работ по проекту и рассмотрены меры по снижению их воздействия на человека. Помимо вредных факторов, влияющих на человека, были также рассмотрены факторы, влияющие на окружающую среду, а именно как образом неправильная утилизация оргтехники влияет на литосферу. А также изучен вопрос о самой вероятной чрезвычайной ситуации во время работы – это пожара. Рассмотрены причины возникновения пожара и действия при его возникновении.

## **6 Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

### **6.1 Введение**

Программные системы являются неотъемлемой частью повседневной жизни человека, так как они охватывают практически все отрасли. Поэтому очень важно создавать качественный продукт. За качество в цикле разработки программного обеспечения отвечает процесс тестирования. Так одной из задач данного проекта является автоматизация процесса тестирования. Автоматизация позволит частично исключить человека из процесса тестирования. Существует большое количество инструментов и технологий для автоматизации, но цель данного проекта применить относительно новую технологию RPA к процессу автоматизации тестирования.

### **6.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **6.2.1 Потенциальные потребители результатов исследования**

В качестве потенциальных потребителей технологии RPA, как инструмента для автоматизации тестирования можно выделить различные компании, которые занимаются разработкой программного обеспечения, ведь тестирования входит в данный цикл. В компаниях тестированием занимается специальный отдел тестирования, который является прямым потребителем рассматриваемой технологии.

Автоматизация тестирования подходит далеко не для всех продуктов и для ее внедрения необходимо анализировать проекты компании. Так если проект рассчитан на долгосрочную работу, то автоматизация будет успешным решением. В основном такие проекты разрабатываются в средних и крупных компаниях, численность которых составляет более 50 сотрудников. Именно такие компании являются главными потребителями технологии.

## 6.2.2 Анализ конкурентных технических решений

В качестве конкурентных решений выделим следующие:

- Ручное тестирование, когда весь процесс тестирования выполняется вручную человеком;
- Автоматизированное тестирование с помощью Selenium, когда часть процесса тестирования, а именно выполнение тес-кейсов автоматизировано и выполняется написанной программой;
- Автоматизированное тестирования с использованием технологии RPA, когда часть процесса тестирования также автоматизировано, но для автоматизации используется другой подход.

Анализ конкурентных технических решений рассчитывается по следующей формуле:

$$K = \sum V_i \cdot B_i,$$

где  $K$  – конкурентоспособность научной разработки,

$V_i$  – вес показателя (в долях единицы),

$B_i$  – балл  $i$ -го показателя.

Анализ проведен с использованием оценочной карты. Результаты проведения анализа представлены в таблице 11, где индексом «р» обозначено ручное тестирование, индексом «s» автоматизированное тестирование с помощью Selenium, а индексом «г» автоматизированное тестирования с использованием технологии RPA.

Таблица 11 – Оценочная карта

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		$B_p$	$B_s$	$B_r$	$V_p$	$V_s$	$V_r$
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Производительность труда	0,15	3	5	5	0,45	0,75	0,75
2. Удобство в эксплуатации	0,1	4	4	5	0,4	0,4	0,5
3. Кроссплатформенность	0,1	5	4	5	0,5	0,4	0,5
4. Простота эксплуатации	0,1	5	2	4	0,5	0,2	0,4

5. Скорость работы	0,15	3	5	5	0,45	0,75	0,75
<b>Экономические критерии оценки эффективности</b>							
1. Конкурен- тоспособность	0,1	4	4	5	0,4	0,4	0,5
2. Стоимость внедрения	0,2	5	4	3	1	0,8	0,6
3. Стоимость поддержки	0,1	4	3	4	0,4	0,3	0,4
<b>Итого</b>	<b>1</b>				<b>4,1</b>	<b>4</b>	<b>4,4</b>

По результатам оценки конкурентных решений можно сказать, что рассматриваемая технология имеет ряд преимуществ таких как скорость работы, а в следствии чего и производительность труда сотрудников, также данная технология удобна в эксплуатации. Но также можно выделить главный недостаток – это стоимость внедрения и поддержки продукта.

### 6.2.3 Технология QuaD

Оценку качества и перспективности можно определить по технологии Qua, которая рассчитывается по следующей формуле:

$$P_{cp} = \sum V_i \cdot B_i,$$

где  $P_{cp}$  – средневзвешенное значение показателя качества и перспективности,

$V_i$  – вес показателя (в долях единицы),

$B_i$  – средневзвешенное значение  $i$ -го показателя.

Анализ проведен с использованием оценочной карты, которая представлена в таблице 12.

Таблица 12 – Оценочная карта

Критерии оценки	Вес критерия	Баллы	Максимальный балл	Относительное значение (3/4)	Средневзвешенное значение (5·2)
1	2	3	4	5	6
<b>Показатели оценки качества разработки</b>					
1. Надежность	0,1	80	100	0,8	0,08
2. Безопасность	0,1	80	100	0,8	0,08
3. Простота эксплуатации	0,15	90	100	0,9	0,135

4. Удобство эксплуатации	0,1	95	100	0,95	0,095
<b>Показатели оценки коммерческого потенциала разработки</b>					
5. Конкурененто-способность	0,1	75	100	0,75	0,075
6. Перспективность	0,1	80	100	0,8	0,08
7. Стоимость внедрения	0,2	40	100	0,4	0,08
8. Стоимость поддержки	0,15	65	100	0,65	0,0975
<b>Итого</b>	<b>1</b>				<b>0,7225</b>

Средневзвешенное значение показателя качества и перспективности получилось 0,7225 это говорит о том, что перспективность данной технологии выше среднего. Это происходит за счет того, что внедрение и поддержка данной технологии требует большого количества затрат, которые нужно хорошо спланировать.

#### 6.2.4 SWOT-анализ

SWOT-анализ применяют для исследования внешней и внутренней среды проекта. На этапе SWOT-анализа необходимо выявить сильные и слабые стороны рассматриваемой технологии, а также рассмотреть ее возможности и угрозы. SWOT-анализ представлен в таблице 13.

Таблица 13 – SWOT-анализ

	<b>Сильные стороны:</b> С1. Снижение трудозатрат специалистов; С2. Скорость выполнения тестов; С3. Более быстрый выпуск продукта; С4. Удобство использования.	<b>Слабые стороны:</b> Сл1. Высокая стоимость внедрения и поддержки; Сл2. Возможно привлечения специалиста по данной технологии; Сл3. Обучение сотрудников в использовании технологии.
<b>Возможности:</b> В1. Автоматизация всего процесса тестирования;	Интеграция с другими системами ведения процесса тестирования	Большие затраты на внедрение, поддержку и обучение сотрудников.

В2. Интеграция с другими системами ведения процесса тестирования; В3. Освобождение времени сотрудников на интеллектуальную деятельность.	позволит автоматизировать весь процесс тестирования; Выпуск продукта в эксплуатацию будет быстрее, за счет скорости выполнения тестов.	
<b>Угрозы:</b> У1. Прекращение поддержки тестов специалистами; У2. Развитая конкуренция на рынке; У3. Несвоевременное финансирование со стороны компании.	За счет удобного и относительно простого интерфейса, продукт выигрывает в конкуренции на данном этапе.	Появление более дешевых средств автоматизации тестирования. Прекращение использования технологии в компании, за счет недостаточных денежных вложений.

В процессе проведения SWOT-анализа выявлены слабые и сильные стороны рассматриваемой технологии, так же определены внешние угрозы и возможности в данной отрасли.

### 6.3 Планирование научно-исследовательских работ

#### 6.3.1 Структура работ в рамках научного исследования

В данной части раздела составлен перечень этапов и работ в рамках проведения научного исследования, проведено распределение исполнителей по видам работ. В качестве исполнителей были выделены: научный руководитель, студент и руководитель от компании, которая предоставила проект для тестирования и спланировала ряд работ. Порядок составления этапов работ, распределение исполнителей по данным видам работ приведен в таблице 14.

Таблица 14 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№	Содержание работ	Исполнитель
Подготовительный этап	1	Определение темы ВКР	Студент
	2	Согласование темы ВКР	Научный руководитель, студент

	3	Сравнительный анализ предметной области	Студент
	4	Анализ используемых технологий автоматизации	Студент
	5	Составление плана тестирования	Руководитель от компании
Основной этап	6	Изучение предоставленного для тестирования проекта	Студент
	7	Составление тест-кейсов для ручного тестирования	Студент
	8	Проведение тестирования по составленным тест-кейсам	Студент
	9	Составление тест-кейсов для автоматизированного тестирования с помощью Selenium	Студент
	10	Проведение тестирования по составленным тест-кейсам	Студент
	11	Подведение итогов тестирования	Научный руководитель, руководитель от компании, студент
	12	Составление тест-кейсов для автоматизированного тестирования с помощью RPA	Студент
	13	Проведение тестирования по составленным тест-кейсам	Студент
Заключительный анализ	14	Анализ по всем видам тестирования	Студент
	15	Подведение итогов и оформление отчета	Научный руководитель, студент

### 6.3.2 Определение трудоемкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоемкости работ каждого из участников научного исследования. Трудоемкость выполнения научного исследования оценивается экспертным

путем в человеко-днях. Для определения ожидаемого значения трудоемкости используется следующая формула:

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5},$$

где  $t_{ож\ i}$  – ожидаемая трудоемкость выполнения  $i$ -ой работы чел.-дн.,

$t_{min\ i}$  – минимально возможная трудоемкость выполнения заданной  $i$ -ой работы чел.-дн.,

$t_{max\ i}$  – максимально возможная трудоемкость выполнения заданной  $i$ -ой работы чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях, учитывающая параллельность выполнения работ несколькими исполнителями по следующей формуле:

$$T_{pi} = \frac{t_{ож\ i}}{ч_i},$$

где  $T_{pi}$  – продолжительность одной работы, раб. дн.,

$t_{ож\ i}$  – ожидаемая трудоемкость выполнения одной работы чел.-дн.,

$ч_i$  – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

### 6.3.3 Разработка графика проведения научного исследования

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{кал},$$

где  $T_{ki}$  – продолжительность выполнения  $i$ -ой работы в календарных днях,

$T_{pi}$  – продолжительность выполнения  $i$ -ой работы в рабочих днях,

$k_{кал}$  – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}},$$

где  $T_{\text{кал}}$  – количество календарных дней в году,

$T_{\text{вых}}$  – количество выходных дней в году,

$T_{\text{пр}}$  – количество праздничных дней в году.

Коэффициент календарности в 2021 году:  $k_{\text{кал}} = \frac{365}{365-66} = 1,22$ .

Все рассчитанные значения необходимо сведены в таблицу 15.

Таблица 15 – Временные показатели проведения научного исследования

№ работы	Исполнители	$t_{\min}$	$t_{\max}$	$t_{\text{ож}}$	$T_p$	$T_k$
1	Студент	1	2	1,4	1,4	1,708 ≈ 2
2	Научный руководитель, студент	1	2	1,4	0,7	0,854 ≈ 1
3	Студент	7	14	9,8	9,8	11,956 ≈ 12
4	Студент	7	14	9,8	9,8	11,956 ≈ 12
5	Руководитель от компании	1	2	1,4	1,4	1,708 ≈ 2
6	Студент	1	2	1,4	1,4	1,708 ≈ 2
7	Студент	2	4	2,8	2,8	3,416 ≈ 3
8	Студент	1	3	1,8	1,8	2,196 ≈ 2
9	Студент	4	7	5,2	5,2	6,344 ≈ 6
10	Студент	1	2	1,4	1,4	1,708 ≈ 2
11	Научный руководитель, руководитель от компании, студент	1	2	1,4	0,47	0,5734 ≈ 1
12	Студент	4	7	5,2	5,2	6,344 ≈ 6
13	Студент	1	2	1,4	1,4	1,708 ≈ 2
14	Студент	1	3	1,8	1,8	2,196 ≈ 2
15	Научный руководитель, студент	1	3	1,8	0,9	1,098 ≈ 1

На основе таблицы 15 строится календарный план-график в виде диаграммы Ганта, представленной на рисунке 23. А на рисунке 24 представлена занятость по исполнителям.

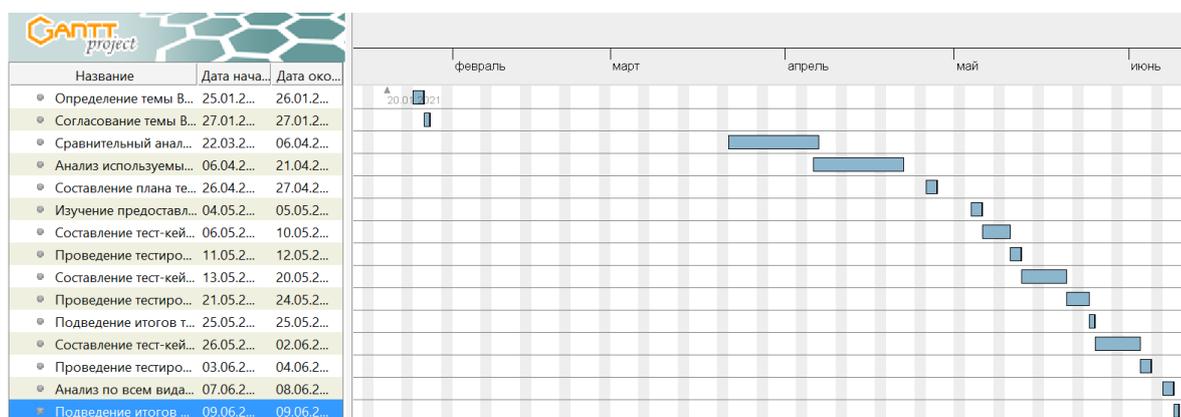


Рисунок 23 – Диаграмма Ганта

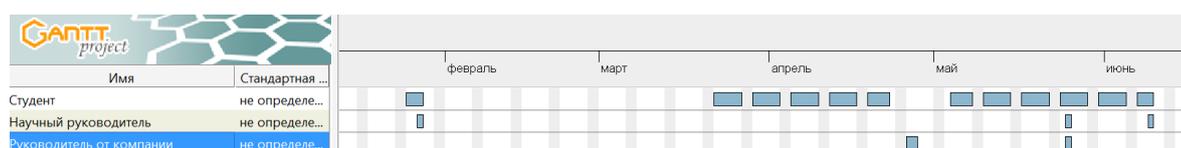


Рисунок 24 – Занятость ресурсов

## 6.4 Бюджет научно-технического исследования (НТИ)

При планировании бюджета НТИ должно быть обеспечено полное и достоверное отражение всех видов расходов, связанных с его выполнением. В процессе формирования бюджета НТИ используется следующая группировка затрат по статьям:

- затраты на специальное оборудование;
- основная заработная плата исполнителей;
- дополнительная заработная плата исполнителей;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

### 6.4.1 Расчет затрат на специальное оборудование

В данную статью включают все затраты, связанные с приобретением специального оборудования, необходимого для проведения работ по конкретной теме. При работе над проектом использовался только уже имеющийся ноутбук. Поэтому его стоимость будет учитываться в виде амортизационных отчислений.

Первоначальная стоимость ноутбука составила 50000 рублей. Срок его полезного использования составляет 3 календарных года. А работа над проектом длилась в течении 3-ех месяцев. Расчет амортизационных отчислений осуществляется по следующей формуле:

$$A = OC_{\text{перв}} \cdot A_{\text{н}},$$

где  $OC_{\text{перв}}$  – первоначальная стоимость,

$A_{\text{н}}$  – норма амортизации.

При этом норма амортизации рассчитывается по формуле:

$$A_{\text{н}} = \frac{1}{n} \cdot 100\%,$$

где  $n$  – срок полезного использования.

Расчет амортизационных отчислений для ноутбука представлен в таблице 16.

Таблица 16 – Расчет амортизационных отчислений

Оборудование	Норма амортизации, %	Амортизационные отчисления, руб.		
		годовые	ежемесячные	суммарные
Ноутбук	33,33	16 500	1 375	4 125

#### 6.4.2 Расчет основной заработной платы

В данную статью включается основная заработная плата студента, научного руководителя и руководителя от компании, непосредственно участвующих в выполнении работ по проекту.

Заработная плата рассчитывается по следующей формуле:

$$Z_{\text{зп}} = Z_{\text{осн}} + Z_{\text{доп}},$$

где  $Z_{\text{осн}}$  – основная заработная плата,

$Z_{\text{доп}}$  – дополнительная заработная плата (12-20% от  $Z_{\text{осн}}$ ).

Основная заработная плата рассчитывается по формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} \cdot T_{\text{р}},$$

где  $Z_{\text{дн}}$  – среднедневная заработная плата работника, руб.,

$T_p$  – продолжительность работ, выполняемых работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}},$$

где  $Z_{\text{м}}$  – месячный должностной оклад исполнителя, руб.,

$M$  – количество месяцев работы без отпуска в течение года:

при отпуске в 24 рабочих дня  $M = 11,2$  месяца, 5–дневная неделя,

при отпуске в 48 рабочих дней  $M = 10,4$  месяца, 6–дневная неделя,

$F_{\text{д}}$  – действительный годовой фонд рабочего времени персонала.

Месячный должностной оклад работника:

$$Z_{\text{м}} = Z_{\text{тс}} \cdot (1 + k_{\text{пр}} + k_{\text{д}}) \cdot k_{\text{р}},$$

где  $Z_{\text{тс}}$  – заработная плата по тарифной ставке, руб.,

$k_{\text{пр}}$  – премиальный коэффициент (0,3),

$k_{\text{д}}$  – коэффициент доплат и надбавок (0,2-0,5),

$k_{\text{р}}$  – районный коэффициент (для Томска – 1,3).

Расчёт баланса рабочего времени приведён в таблице 17.

Таблица 17 – Баланс рабочего времени

<b>Показатель рабочего времени</b>	<b>Дни</b>
Календарные дни	365
Нерабочие дни (праздники/выходные)	66
Потери рабочего времени (отпуск/невыходы по болезни)	56
Действительный годовой фонд рабочего времени	243

Результат расчёта основной заработной платы приведён в таблице 18.

Зарплата студента принимается за 10000 рублей по тарифной ставке, зарплата научного руководителя за 30000 рублей, зарплата руководителя от компании 30000 рублей.

Таблица 18 – Расчет основной заработной платы

<b>Исполнитель</b>	$Z_{\text{тр}}$	$k_{\text{пр}}$	$k_{\text{д}}$	$k_{\text{р}}$	$Z_{\text{м}}$	$Z_{\text{дн}}$	$T_p$	$Z_{\text{осн}}$
Студент	10000	0,3	0,2	1,3	19500	834,6	54	45 068,4

Научный руководитель	30000	0,3	0,2	1,3	58500	2 503,7	3	7 511,1
Руководитель от компании	30000	0,3	0,2	1,3	58500	2 696,3	3	8 088,9
<b>Итого</b>								<b>60 668,4</b>

### 6.4.3 Расчет дополнительной заработной платы

Затраты по дополнительной заработной плате исполнителей учитывают величину доплат за отклонение от нормальных условий труда, а также выплат, связанных с обеспечением гарантий и компенсаций.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}},$$

где  $k_{\text{доп}}$  – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,12 – 0,15).

Расчет дополнительной заработной платы представлен в таблице 19.

Таблица 19 – Расчет дополнительной заработной платы

Исполнитель	$Z_{\text{осн}}$	$k_{\text{доп}}$	$Z_{\text{доп}}$
Студент	45 068,4	0,13	5 858,9
Научный руководитель	7 511,1		976,4
Руководитель от компании	8 088,9		1 051,6
<b>Итого</b>			<b>7 886,9</b>

### 6.4.4 Расчет отчислений во внебюджетные фонды

Данная статья включает обязательные отчисления в фонд социального страхования, пенсионный фонд и фонд медицинского страхования. Размер отчислений зависит от размера заработной платы по следующей формуле:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}}),$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

Расчет отчислений во внебюджетные фонды представлен в таблице 20.

Таблица 20 – Расчет отчислений во внебюджетные фонды

<b>Исполнитель</b>	$Z_{\text{осн}}$	$Z_{\text{доп}}$	$k_{\text{внеб}}$	$Z_{\text{внеб}}$
Студент	45 068,4	5 858,9	0,302	15 380
Научный руководитель	7 511,1	976,4		2 563,2
Руководитель от компании	8 088,9	1 051,6		2 760,4
<b>Итого</b>				<b>20 703,6</b>

#### 6.4.5 Расчет накладных расходов

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, почтовые и телеграфные расходы, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$Z_{\text{накл}} = k_{\text{нр}} \cdot \text{сумма статей (1 – 4)},$$

где  $k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы.

Расчет накладных расходов представлен в таблице 21.

Таблица 21 – Расчет накладных расходов

<b>Статьи затрат</b>	<b>Сумма, руб.</b>
Затраты на специальное оборудование	4 125
Основная заработная плата	60 668,4
Дополнительная заработная плата	7 886,9
Отчисления во внебюджетные фонды	20 703,6
Коэффициент, учитывающий накладные расходы	0,16
<b>Накладные расходы</b>	<b>14 941,4</b>

#### **6.4.6 Формирование бюджета затрат научно-исследовательского проекта**

Рассчитанная величина затрат научно-исследовательской работы является основой для формирования бюджета затрат проекта.

Определение бюджета затрат на научно-исследовательский проекта приведено в таблице 22.

Таблица 22 – Расчет бюджета затрат НИИ

<b>Наименование статьи</b>	<b>Сумма, руб.</b>	<b>Примечание</b>
Затраты на специальное оборудование	4 125	См. раздел 6.4.1
Основная заработная плата	60 668,4	См. раздел 6.4.2
Дополнительная заработная плата	7 886,9	См. раздел 6.4.3
Отчисления во внебюджетные фонды	20 703,6	См. раздел 6.4.4
Накладные расходы	14 941,4	См. раздел 6.4.5
<b>Итого</b>	<b>108 325,3</b>	<b>Сумма статей 1-5</b>

#### **6.5 Вывод по разделу**

В ходе написания раздела были определены потенциальные потребители решения, проведен анализ конкурентных технических решений с помощью оценочной карты, а также был проведен анализ по технологии QuaD, который оценивает перспективность решения. Составлен SWOT-анализ с выявлением сильных и слабых сторон проекта, а также возможностей и угроз.

Был составлен перечень этапов и работ в рамках проведения научного исследования, проведено распределение исполнителей по видам работ. По полученным видам работ была определена трудоемкость и с помощью диаграммы Ганта выполнено планирование работ. Кроме того, был определен бюджет проекта, а именно были рассчитаны затраты на специальное оборудование, затраты на основную и дополнительную заработные платы исполнителей, затраты на отчисления во внебюджетные фонды и налоговые расходы.

## Заключение

В рамках подготовки и написания выпускной квалификационной работы был проведен полный процесс тестирования программного обеспечения от анализа системы до анализа результатов выполнения тестовых сценариев. Были изучены различные методы для разработки тестовых сценариев и проведения тестирования. Эти методы подразделяются на два вида: ручное и автоматизированное тестирования. Так были разработаны тестовые сценарии для ручного тестирования, а также проведено выполнение разработанных тестов вручную. Помимо ручного тестирования было проведено автоматизированное на двух различных платформах, таких как: Selenium, который предполагает написание скриптов на языке программирования (был выбран Python) и технология RPA с использованием платформы UiPath, которая позволяет разрабатывать тестовые сценарии в виде блок-схем.

По результатам проведения трех различных циклов тестирования были рассчитаны показатели эффективности от автоматизации на Selenium и на UiPath. А также просчитаны временные затраты на тестирования. Результаты показывают, что эффективность от автоматизации наступает не сразу, а только спустя какое-то количество выпусков системы. Исходя из данной работы автоматизация RPA на начальном этапе выигрывает у автоматизации на Selenium, но был замечен факт того, что спустя некоторое количество выпуском (в данном случае 23) автоматизация на Selenium становится эффективнее. Отсюда можно сделать вывод, что автоматизация на Selenium эффективна для долгосрочных систем, предполагающих большое количество выпусков в процессе разработки, а RPA больше подходит для систем, имеющих небольшое количество выпусков (примерно от 5-10). Но для совсем небольших систем автоматизация будет не эффективна и лучше использовать ручное тестирование.

## Список используемых источников

1. «Трудовой кодекс Российской Федерации» от 30.12.2001 N 192-ФЗ (ред. от 30.04.2021) (с изм. и доп., вступ. в силу с 01.05.2021) [Электронный ресурс] / КонсультантПлюс – URL: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_34683/](http://www.consultant.ru/document/cons_doc_LAW_34683/), свободный. Дата обращения: 30.05.2021
2. ГОСТ 12.2.032-78 Рабочее место при выполнении работ сидя. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200003913>, свободный. Дата обращения: 30.05.2021
3. ГОСТ 21889-76 Система "Человек-машина". Кресло человека-оператора. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200012832>, свободный. Дата обращения: 30.05.2021
4. ГОСТ 22269-76 Рабочее место оператора. Взаимное расположение элементов рабочего места. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200012834>, свободный. Дата обращения: 30.05.2021
5. ГОСТ Р ИСО 9241-4-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов (VDT). Часть 4. Требования к клавиатуре. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL: <https://docs.cntd.ru/document/1200076334>, свободный. Дата обращения: 30.05.2021
6. ГОСТ Р ИСО 9241-5-2009. Эргономические требования к проведению офисных работ с использованием видеодисплейных терминалов

(VDT). Часть 5. Требования к расположению рабочей станции и осанке оператора. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/1200076558>, свободный. Дата обращения: 30.05.2021

7. ГОСТ 12.1.003-2014 «Шум. Общие требования безопасности». Общие требования безопасности труда. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/1200118606>, свободный. Дата обращения: 30.05.2021

8. СП 51.13330.2011. Защита от шума. Актуализированная редакция СНиП 23-03-2003. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/1200084097>, свободный. Дата обращения: 30.05.2021

9. СП 52.13330.2016 Естественное и искусственное освещение [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/456054197>, свободный. Дата обращения: 30.05.2021

10. ГОСТ 12.1.006-84 ССБТ. Электромагнитные поля радиочастот. Общие требования безопасности. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/5200272>, свободный. Дата обращения: 30.05.2021

11. ГОСТ 12.1.005-88 ССБТ. Общие санитарно-гигиенические требования к воздуху рабочей зоны. [Электронный ресурс] / Электронный фонд правовых и нормативно-технических документов – URL:

<https://docs.cntd.ru/document/1200003608>, свободный. Дата обращения: 30.05.2021

12. Selenium with Python [Электронный ресурс] / URL: <https://selenium-python.readthedocs.io/index.html>, свободный. – Дата обращения: 16.04.2021.
13. Автоматизированное тестирование сайта – за и против. [Электронный ресурс] / URL: <https://www.uplab.ru/blog/automated-testing-of-website/>. – Дата обращения: 24.04.2021.
14. Роман Савин. Тестирование dot com или пособие по жесткому обращению с багами в интернет-стартапах. / Издательство «Дело», Москва, 2007. – 315 с.
15. Святослав Куликов. Тестирование программного обеспечения. Базовый курс. / Издательство «Четыре четверти», Минск, 2017. – 314 с.
16. UiPath Academy [Электронный ресурс] / URL: <https://academy.uipath.com/static-page/5>. – Дата обращения: 02.06.2021.
17. Гребенюк Виктор. Оценка целесообразности внедрения автоматизированного тестирования. / Интернет-журнал «НАУКОВЕДЕНИЕ» №1 2013 / URL: <https://naukovedenie.ru/PDF/13tvn113.pdf>. – Дата обращения: 08.06.2021.